D.JRA.6.1.1

State-of-the-art with regards to user-perceived Quality of Service and quality feedback

Deliverable version No: 1.0

Sending date: 31/05-2004

Lead contractor N°: 49

Dissemination level: Public

URL reference of the workpackage: www.eurongi.org/...

Project acronym: Euro-NGI.
Project full title: Design and Engineering of the Next Generation Internet, towards convergent multi-service networks.
Type of contract: NETWORK OF EXCELLENCE.
Contract N°.: 507613.
URL's project: http://www.eurongi.org

Editor's name: Markus Fiedler

Editor's e-mail address: markus.fiedler@bth.se

Partner Number	Partner Name	Contributor Name	Contributor e-mail address
3	UniVie	H. Hlavacs	helmut.hlavacs@univie.ac.at
4	JKU	G. Kotsis	gabriele.kotsis@jku.ac.at
4	JKU	T. Grill	thomas.grill@jku.ac.at
10	IRISA	S. Mohamed	samir.mohamed@irisa.fr
10	IRISA	G. Rubino	gerardo.rubino@irisa.fr
10	IRISA	M. Varela	martin.varela@irisa.fr
10	INRIA	V. Ramos	victor.ramos@sophia.inria.fr
10	INRIA	C. Bakarat	chadi.bakarat@sophia.inria.fr
10	INRIA	E. Altman	altman@sophia.inria.fr
17	UniWue	K. Tutschku	tutschku@informatik.uni-wuerzburg.de
22	RC-AUEB	M. Dramitinos	mdramit@aueb.gr
22	RC-AUEB	G.D. Stamoulis	gstamoul@aueb.gr
22	RC-AUEB	C. Courcoubetis	courcou@aueb.gr
33	Telenor	T. Jensen	terje.jensen1@telenor.com
49	BTH	M. Fiedler	markus.fiedler@bth.se
49	BTH	P. Carlsson	patrik.carlsson@bth.se
56	UP	H. de Meer	demeer@fmi.uni-passau.de

Project acronym: Euro-NGI.
Project full title: Design and Engineering of the Next Generation Internet, towards convergent multi-service networks.
Type of contract: NETWORK OF EXCELLENCE.
Contract N°.: 507613.
URL's project: http://www.eurongi.org

Abstract

This deliverable D.JRA.6.1.1 presents a review of the state-of-the-art with regards to Quality of Service from the user's perspective and quality feedback, which is the topic of the corresponding work package WP.JRA.6.1 as part of the Joint Research Activity 6 "Socio-Economic Aspects of Next Generation Internet" of the Network of Excellence "Euro-NGI". The document contains a survey of Quality of Service-related standards and discusses the current status regarding Quality of Servce in the Internet. The central role of the user is highlighted, and methods how to relate user perception to technical parameters on application and network level are discussed. Furthermore, currently existing quality feedback and management facilities in Internet are reviewed. Complementary work of the involved partners within these fields is presented, showing the broad range of compentence of the partners within the scope of JRA.6.1. Finally, relevant research issues are identified, providing a promising basis for future joint research.

Contents

Chapter 1

Introduction

Thanks to the advent of new services and advances in communications research and development, the Internet has shown its potential to penetrate almost all aspects of life. Recently, many traditional, ineffective and expensive public and private services have got so-called e-services associated with them intended to take over customers in the long run. Also, personal communication (telephony; messaging; etc.) and entertainment (streaming; gaming; filesharing; etc.) is increasingly carried out via the Internet. Thus, the tastes of Next Generation Internet are clearly intended to improve Quality of Life through networked, user-oriented, personalized services generating added value and revenue for users and providers, respectively.

To make these value chains work, it is required that the services behave as expected by the users (men, machines, systems). In other words, a certain *Quality of Service* (QoS) has to be met in terms of speed, accuracy and reliability [1]. If such expectations are not met, there might be different kinds of consequences: Processes may hang or become instable; people may get impatient or angry. In the end, a service might not be considered be of any value to a user and be abandonned, which may lead to loss of revenue for service, content and network providers. No matter whether their origin is found in the application or in the network, perceived quality problems might lead to acceptance problems especially if money is involved [2].

Thus, the introduction of new, challenging services can neither leave perceived quality nor pricing out of scope. The user should be satisfied with the perceived quality and feel the pricing of the service to be fair. The degree of satisfaction, i.e. the subjective quality, is influenced by the technical, objective quality stemming from the application and the interconnecting network(s). For this reason, subjective quality as perceived by the network has to be linked to objective, measurable quality, which is expressed in application and network performance parameters. The latter represent the interface to network-centric research dealing with architectures, dimensioning, resource allocation, routing, optimization, measurement and modelling by providing target values for parameters and possibilities to carry out experiments, which is illustrated by Figure 1.1.

At the same time, proper quality management involving users, providers, applications and networks is needed. The key to this kind of control is quality feedback between these entities, which will be surveyed and developed further. Improved quality management

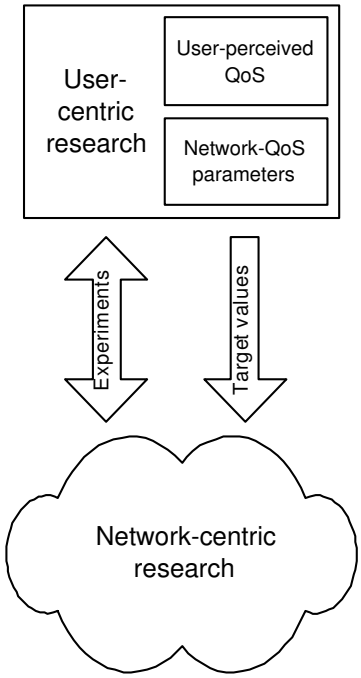


Figure 1.1: Interaction of user-centric and network-centric research

paradigms will influence the development of both services and network management, respectively.

Within the Network of Excellence "Euro-NGI", a group of partners has gathered around these issues within the work package JRA.6.1 "Quality of Service from the user's perspective and feed-back mechanisms for quality control", where JRA stands for Joint Research Activity. The scope of the first deliverable D.JRA.6.1.1 is to provide a view of the state-of-the-art of user-perceived QoS and quality feedback in the Internet as exemplified by the work of these partners. D.JRA.6.1.1 is intended to be a starting-point for further joint research work within the scope of user-perceived QoS and related quality management within JRA.6.1 and in collaboration with other JRAs.

The remainder of this deliverable is stuctured as follows: Chapter 2 discusses QoS from the viewpoint of standards, Internet, user and management. Chapter 3 presents complementary views and results from the partners involved in this work package. Finally, Chapter 4 draws conclusions indicating directions for future work.

Chapter 2

Quality of Service

The notion of *Quality of Service* (QoS) is central to this work package and its deliverables, which motivates the need for reviewing the corresponding terms and actors as well as the relationships between them. Section 2.1 reviews some important standards with regards to QoS, while section 2.2 discusses the current situation in best-effort Internet. Section 2.3 discerns between user-perceived quality from application- and network-level quality, and Section 2.4 presents existing quality feed-back mechanisms as part of concurrent quality management.

2.1 Quality of Service-related standards

2.1.1 ITU / ISO

The International Telecommunication Union (ITU)[1] has created a set of recommendations in the area of QoS. Many of theses recommendations have been published also by the International Standardization Organization (ISO).[2] The recommendations cover many different areas on the field of general QoS frameworks, QoS management and measurement, QoS seen from the user and QoS related to multmedia applications.

ITU-T E.800 A thorough survey of the QoS concept is found in the ITU-T standard E.800 [3] from 1994 relating QoS and network performance and providing a set of performance measures especially for telecommunication networks. QoS is defined as "the collective effect of service performance which determine the degree of satisfaction of a user of the service". It comprises (see Figure 1/E.800):

- Service support performance;
- Service operability performance;

[1] http://www.itu.ch
[2] http://www.iso.org

- Serveability, including service accessibility, retainablity and integrity performance;
- Service security performance.

Serveability on the QoS side interfaces with trafficability performance on the network performance side, addressing resources and facilities, dependability, and transmission performance. *Network performance* is defined as "the ability of a network or network portion to provide the functions related to communications between users". Thus, the framework provides clear links between user satisfaction (termed QoS) and network performance parameters such as availability, mean time to failure, mean down time etc. The standard defines a great amount of parameters related to telephony-type networks are defined. However, no quantitative target values, called QoS objectives, are provided.

ITU-T E.860 The basis formed by E.800 is extended in the ITU-T standard E.860 [1] from 2002 forming a framework for a *Service Level Agreement* (SLA). It is argued that, in face of growing competition, QoS becomes a distinctive property of a service or network provider, while another challenge is the increasing demand of services involving several providers and different kinds of network technologies. An SLA provides means to formalize the relationhips between a provider (delivering a service) and a user (receiving a service); it is "a formal agreement between two or more entities that is reached after a negotiating activity with the scope to assess service characteristics, responsibilities and priorities of every part". The recommended structure of an SLA is shown in Figure 2.1. The introduction defines the purpose of the SLA (e.g. defining service levels for customer's satisfcation), while the scope reflects the services of interest and their target performance. Confidentiality agreements might be necessary with respect to competitors.

In [1], QoS is defined as the "degree of conformance of the service delivered to a user by a provider in accordance with an agreement between them". The quality of the service function is valued in three criteria [4]:

- *Speed* = aspects of temporal efficiency of a function, defined on measurements made on sets of time intervals, e.g. delays;
- *Accuracy* = degree of correctness, based on ratio or rate of incorrect realizations of a function, e.g. losses;
- *Reliability* = degree of certainty with which a function is performed, which is related to dependability.

In other words, QoS is a measurable good with a market value that is always related to the corresponding user's perception. Such a user (or customer) can be an end user, a regulatory entity or another service provider (SP).

The QoS agreement shown in Figure 2.1 is also called *Service Quality Agreement* (SQA). While the business interface deals with negotiation, reporting and reaction issues, the technical interface exchanges service-specific information and allows for measurements as a basis for deriving QoS parameters directly or indirectly, i.e. as functions of other direct parameters. Knowledge and understanding of traffic patterns is important at the

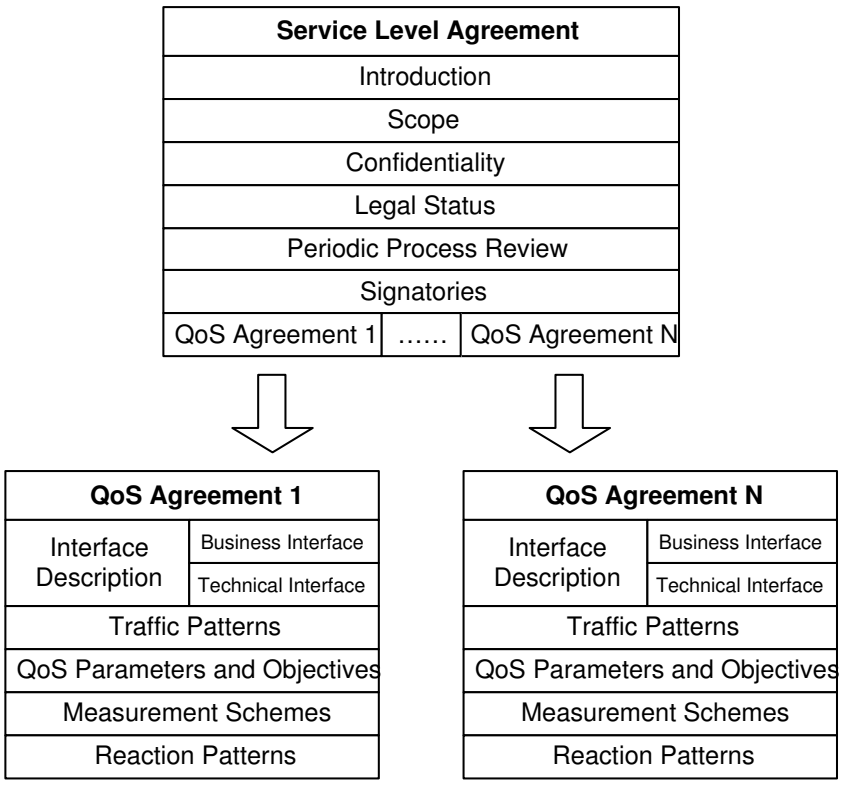


Figure 2.1: Generic structure of a Service Level Agreement [1].

interface between providers, and reaction patterns may be needed in case of deviations or violations, e.g. [1]

- provider's reaction to an incoming traffic that differs from the description in the SLA;
- user's behaviour when service provider does not provide QoS agreed in the SLA.

Such reaction patters include [1]

- no action;
- monitoring the achieved QoS;
- traffic flow policing through traffic shaping and/or admission control;
- reallocating resources;
- warning signals to customer/SP when thresholds are being crossed;
- suspending or aborting the service.

Some QoS parameters depend on specific services, others are service-independent. Furthermore, different timelines might apply (Service: decades/years; user: years/months;

session: hours/minutes). However, the parameters should be well-understood by the involved parties. Their objectives might be given by target values, thresholds or ranges, and the matching might be expressed by *Service Degradation Factors* (SDF). Measurement specifications should refer to "what, when, where and who" (but not necessarily "how") and may include the methodology to evaluate measurement results.

Especially in multi-provider environments, the *one stop responsability* concept [5] is desirable from the viewpoint of the end user: Instead of having to deal with many providers amd correponding SLAs, there is one primary provider responsible for fulfilling the SLA, while the sub-providers are hidden. The primary provider might apply the same one stop reponsibility to its sub-providers. The result is a chain of SLAs. This concept is important for the provisioning of *End-to-End QoS*. In this case, it might be interesting to also negotiate an *End-to-End SLA*; details are found in [1].

Section 3.1 provides a detailed view on these issues.

ITU-T X.140 The ITU-T Recommendation X.140 [6] comprises a general framework for user-oriented QoS measures in data networks. The described parameters are valid for circuit switched and packet switched networks. QoS parameters for circuit switched networks can be found, for instance, in ITU-T Rec. X-130 and X.131, those for packet switched networks for instance in ITU-T Rec. X.134 to X.137.

Table 2.1: General QoS parameters for communication via public data networks (ITU-T Rec. X.140).

Function / Criterion	Speed	Accuracy	Dependability
Access	Access delay	Incorrect access prob.	Access denial prob.
User inform. transfer	-UI transfer delay -UI transfer rate	-UI error prob. -Extra UI delivery prob. -UI misdelivery prob.	UI loss prob.
Disengagement	Diseng. delay	Diseng. denial prob.	

The parameters are shown in Table 2.1. They describe the QoS during normal hours of service operation and the frequency and duration of service outages. As described in the recommendation, a *block* is a basic unit of user information (UI) that is transferred over the network [Sei94]. This can be a web page, a video frame or a transferred file. The following list contains an explanation of the parameter categories:

- Access Delay. The time elapsed between an access request and successful access. This parameter is generalized to *response time* as the time between manually issuing a request to the system, until the request is satisfied.
- Incorrect Access Probability: The ratio of total access attempty that result in incorrect access to total access attempts in a specified sample.

- Access Denial Probability. The probability that a request is denied and the user is notified.
- User Information Transfer Delay. The latency of a block sent over the network.
- User Information Transfer Rate. The throughput experienced when transferring a block.
- User Information Error Probability. The probability for bit errors or bit losses occurring in a transferred block.
- Extra User Information Delivery Probability. The ratio of total (unrequested) extra blocks to total blocks by a destination user in a specified sample.
- User Information Misdelivery Probability. The ratio of total misdelivered user blocks to total user blocks between a specified source and destination user in a specified sample.
- User Information Loss Probability. This is the probability that a block is lost during transfer.
- Disengagement Delay. This is the elapsed time between the attempt to close a connection until the connection is actually closed.
- Disengagement Denial Probability. The ratio of total disengagement attempts that result in disengagement denial to total disengagement attempts in a specified sample.

ITU-T X.641 / ISO 13236 The ITU-T Rec. X.641 has also been published as ISO 13236 standard. It contains a general framework for describing the QoS of distributed systems. The framework defines and explaines general terms and concepts about distributed objects that interact with each other.

The concept of this framework is as follows. The basic starting point are the *services* provided by objects of the system. When accessing such a service, a client may observe *QoS characteristics* of the system, which denotes some aspect of the QoS of a system that can be identified and measured.

The goal of the system is to yield what is defined by user *QoS requirements*, which are quantified and expressed by *QoS requirements*. These QoS requirements can be expressed as QoS parameters, which may include

- a desired level of characteristic,
- a maximum or minimum level of characteristic,
- a measured value,
- a threshold level,
- a warning or signal to take corrective action, or

- a request for operations on managed objects relating to QoS, or the results of such operations.

The QoS of a sytem is managed by *QoS management functions*, which may include

- establishment of QoS for a set of QoS characteristics,
- monitoring of the observed values of QoS,
- maintenance of the actual QoS as close as possible to the target QoS,
- control of QoS targets,
- alerts as a result of some event relating to QoS management.

When measuring QoS characteristics, the measured values may be of several types. A *generic characteristic* denotes a characteristic which is independent of what it is applied to later, for instance *time delay*. A *specialization* of such a generic characterisation denotes the generic characterisation applied to a specific measurement target, for instance *transit delay*, a further specialization would define for instance *transit delay between two hosts*. A *derived characteristic* is a statistic of specializations, for instance the mean, variance or minimum.

The recommendation then describes generic mechanisms for QoS management, which include

- A QoS prediction phase, where the QoS that will be observed is predicted.
- An establishment phase, where the QoS is agreed on and established.
- The operational phase, where the QoS is monitored.

ITU-T X.642 In the ITU-T Recommendation X.642 [7] an overview over ITU Recmmendations and other standards from ISO and IETF related to QoS is given. Table 2.1 contains a subsample of this overview, consisting of the recommendatio/standard sources and general categories.

This recommendation also defines general QoS mechanisms for predicting, negotiating, agreeing and establishing QoS for unicast and multicast applications.

2.1.2 IETF

For the Internet Engineering Task Force (IETF), QoS is primarily a question of routing packets through a network. Concequently, the QoS related standards of the IETF focus on network management and routing mechanisms. RFCs related to QoS are given in Table 2.3.

Table 2.2: QoS related recommendations and standards (ITU-T Rec. X.642).

Source	Category	Subcategory
ITU-T/ISO	QoS for lower layers	Service definitions
		Generalized protocol specifications
		Protocol specifications for specific technologies
	QoS for upper layers	OSI higher layers
		Message handling systems (MHS)
		OSI system management suporting QoS management
	QoS for Open Distributed Systems	
ISO/IEC only	Intenational Standardized Profiles (ISPs)	
ITU-T only	G-Series	Transmission systems and media, digital systems and networks
	I-Series	Integrated Services Digital Networks (ISDNs)
	X-Series	Data networks and open system communication
IETF	IntServ, DiffServ, IPv6, RTP, RSVP, ...	

Table 2.3: QoS related RFCs.

QoS Mechanism	RFCs
IntServ	1633, 1819, 1821, 1883, 1889, 2205 - 2216
IPv6	1883
RSVP	2205, 2210, 2211, 2212
DiffServ	2474, 2475, 2597, 3246, 3247, 2697, 2698, 2963, 2983, 3260, 3289, 3290

2.2 Quality of Service in the Internet

2.2.1 The Internet Paradigm

During the 1990's, applications have become increasingly reliant on the use of the Internet protocols to provide data communications facilities. The use of the Internet protocols seems likely to increase at an extremely rapid rate and the Internet Protocol (IP) will be the dominant data communications protocol in the next decade. IP is being used for a huge variety of "traditional" applications, including e-mail, file transfer and other general non-real-time communication. However, IP is now being used for real-time applications that have QoS-sensitive data flows. A flow is a stream of semantically related packets which may have special QoS requirements, e.g. an audio stream or a video stream. Applications such as conferencing (many-to-many communication based on IP multicast), telephony – voice-over-IP (VoIP) – as well as streaming audio and video are being developed using Internet protocols.

The Internet was never designed to cope with (such) a sophisticated demand for services [8]. Today's Internet is built upon many different underlying network technologies, of different age, capability and complexity. Most of these technologies are unable to cope with such QoS demands. Also, the Internet protocols themselves are not designed to support the wide range of QoS profiles required by the huge plethora of current (and future) applications.

Let us first examine the service that IP offers. IP offers a connectionless datagram service, giving no guarantees with respect to delivery of data: no assumptions can be made about the delay, jitter or loss that any individual IP datagrams may experience. As IP is a connectionless, datagram service, it does not have the notion of flows of datagrams, where many datagrams form a sequence that has some meaning to an applications. For example, an audio application may take 40 ms "time-slices" of audio and send them in individual datagrams. The correct sequence and timeliness of datagrams has meaning to the application, but the IP network treats them as individual datagrams with no relationship between them. There is no signalling at the IP-level: there is no way to inform the network that it is about to receive traffic with particular handling requirements and no way for IP to tell or signal users to back-off when there is congestion.

At IP routers, the forwarding of individual datagrams is based on forwarding tables using simple metrics and (network) destination addresses. There is no examination of the type of traffic that each datagram may contain - all data is treated with equal priority. There is no recognition of datagrams that may be carrying data that is sensitive to delay or loss, such as audio and video.

One of the goals of IP was to be robust to network failure. That is why it is a datagram-based system that uses dynamic routing to change network paths in event of router overloads or router failures. This means that there are no fixed paths through the network. It is possible that during a communication session, the IP packets for that session may traverse different network paths. The absence of a fixed path for traffic means that, in practice, it can not be guaranteed that the QoS offered through the network will remain consistent during a communication session. Even if the path does remain stable, because

IP is a totally connectionless datagram traffic, there is no protection of the packets of one flow, from the packets of another. So, the traffic patterns of a particular user's traffic affects traffic of other users that share some or all of the same network path (and perhaps even traffic that does not share the same network path!).

At the individual routers, the process of forwarding a packet involves, taking an incoming packet, evaluating its forwarding path, and then sending it to the correct output queue. Packets in output queues are serviced in a simple first-come first-serve (FCFS) order, i.e. the packet at the front of the queue is transmitted first. The ordering of packets for transmission takes the general term on scheduling, and we can see FCFS is a very simple scheduling mechanism. FCFS assumes that all packets have equal priority. However, there is a strong case to instruct the router to give some traffic higher priority than other traffic. For example, it would be useful to give priority to traffic carrying real-time video or voice. How do we distinguish such priority traffic from non-priority traffic, such as, say e-mail traffic. The IPv4 type of service (ToS) do offer a very rudimentary form of marking traffic, but the semantics of the ToS markings are not very well defined. Subsequently, the ToS field is not widely used across the Internet. However, it can be used effectively across corporate Intranets.

2.2.2 Internet Service Providers

The network layer is often assumed to be an autonomous system of an *Internet Service Provider* (ISP). Though this is a meaningful level of abstraction, in order to avoid the large amount of technical details regarding the network infrastructure, we briefly comment on the major entities of the network level. In practice, the Internet network infrastructure is composed of a large number of interconnecting networks. Interconnection is the means by which customers can connect to different network providers and still receive end-to-end service that spans two or more networks. The idea is that the service provided to a customer of one given network can use the infrastructures of a number of other network providers.

Peering agreements have some distinct characteristics. Peering partners only exchange traffic on a bilateral basis that originates from customers of one partner and terminates to customers of the other partner. This implies that customers of one network can send or receive information from customers of the other network. A peering partner does not act as an intermediary that accepts traffic from one partner and transits this traffic to another partner. Peering traffic is exchanged on a settlement-free basis also known as "sender-keeps-all". The only costs involved in peering are the purchase of equipment and the provision of transmission capacity needed for each partner to connect to some common traffic exchange point. It is interesting that peering agreements do not specify any minimum performance on the way a network may handle traffic originating from a peer, which is usually handled as "best-effort". Network providers consider several factors when negotiating peering agreements. These include the customer base of their prospective peer and the capacity and span of the peer's network. Clearly, some providers have greater bargaining power than others. It may be of no advantage for a provider with a large customer base to peer on an equal basis with a provider with a small customer base. *Transit agreements* are the other type of interconnection agreements. There is an important difference

between peering and transit. Using transit one partner pays another partner for interconnection and therefore becomes a customer. The partner selling transit services will route traffic from the transit customer to its own peering partners as well as to other customers. In this case this intermediate network provides a clearly defined transport service for the transit traffic of the first network, and hence can charge for it in a way that reflects the service contract and the actual usage.

The Internet connectivity market is structured hierarchically, comprising three main levels of participants: end-users, ISPs and *Internet Backbone Providers* (IBPs). End-users are at the bottom of the hierarchy and access the Internet via ISPs. End-users include individual and business customers. At the top of the hierarchy, IBPs own the high speed and high capacity networks which provide global access and interconnectivity. They primarily sell wholesale Internet connectivity services to ISPs. ISPs then resell connectivity services, or add value and sell new services to their customers. However, IBPs may also become involved in ISP business activities by selling retail Internet connectivity services to end-users. Two markets are identified in the Internet connectivity value chain: the wholesale market, and the retail for global access and connectivity to end-users. There are two main types of contracts in terms of pricing: between end-users and ISPs for primary Internet access and between ISPs and IBPs for interconnection. In the early days, when the Internet was serving exclusively the public sector mainly for research and education purposes, interconnection was a public good and its provision was organized outside competitive markets. Today interconnection is primarily commercial, yet its basic architectures remain unchanged. Network externalities generate powerful incentives for interconnection.

2.2.3 Summary

From the two preceding sections, it is seen that Internet service is basically best-effort all the way between sender and receiver. Overdimensioning is still the way of keeping QoS-related problems small; approaches like IntServ or DiffServ (cf. Section 2.1.2) are not operational. Signalling happens implicitly through packet delay and loss, which is measured by some end-to-end protocols (TCP or RTP) and used for the purpose of end-to-end control. Section 2.4 discusses such feed-back solutions in greater detail. Section 3.2 proposes some enhancements of the basic Internet service in order to improve QoS support.

The signification and contents of SLAs are still unknown to most users; however, Section 3.1.3 reports a joint project between a regulatory authority and a telecom users association.

2.3 Quality of Service from the user's perspective

2.3.1 Different kinds of QoS

Due to the very nature of communication following the OSI model, in which each layer provides service to the upper layer(s), we have to distinguish several levels of QoS, see

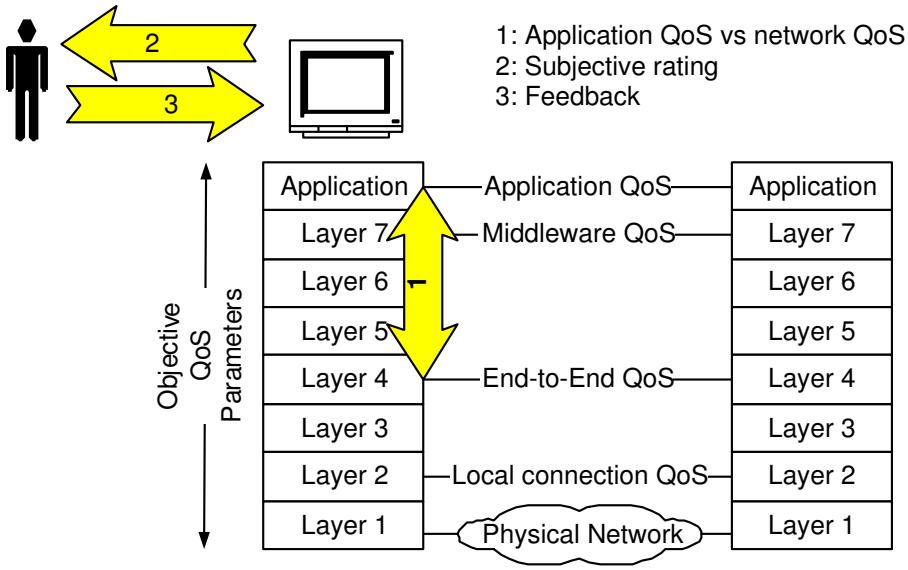


Figure 2.2: Network stack influencing the perceived QoS; original figure from [9].

Figure 2.2.

- On the transport-oriented levels 1 to 4, end-to-end QoS – or simply *network QoS* – is determined by the conditions on physical, link and network level and by the transport protocol itself.
- The application-oriented levels 5 to 7 perceive the end-to-end QoS and turn it into middleware QoS.
- This middleware QoS is perceived by the application, which in turn acts upon this and makes the user experience the *application QoS*.

The user does not experience network problems such as delays, losses, etc. directly but through the application in use. In classical telephony, on the other hand, one may even able to hear problems on the physical level (bit errors leading to short drop-outs; impedance problems leading to echo; etc.). However, given the complexity and reactiveness of applications and protocols, it is very important to distinguish between application QoS and network QoS (arrow "1" in Figure 2.2) where the actual communication provisioning happens. Problems perceived by a user might have their origins in the application instead of the network, while on the other hand, the effects of network problems might be damped by the application such that the user does not feel any disturbance at all. However, the user will rate the application and thus also the network (arrow "2"); perceived connectivity problems are quite often blamed to the latter. It is thus important to correlate what is happening both in network and application to the user experience in order to work on the right problems. Moreover, in case of quality problems, user and/or provider will react in some way (arrow "3"), which is detailed in Section 2.4.

As pointed out before, both application and network stacks can cause troublesome behavior. The user perceives the overall result, no matter of where the very problem is located.

2.3.2 User perception and rating

Depending on the task a user is carrying out, problems with networks or applications are felt to be more or less annoying [2]. Users rate the application QoS (and thus also the network QoS) in a subjective and individual way depending on the usability that is perceived, which is discussed in detail in Section 3.3. User satisfaction typically depends on perceived response times [2], on the user's own expectations and also on the pricing model [10].

The very user rating happens either explicitly (by commenting, complaining, etc.) or implicitly (by being dissapointed, giving up using the service, etc.) upon passing of certain acceptance thresholds. For a service provider, it is important to find out about such thresholds and their correlation with problematic states of applications and/or networks. *Utility Curves* (UC) provide a formal technique to directly relate network state, such as available bandwidth, to end-user perceived QoS. Section 2.3.5 discusses the the concept of utility functions in greater detail. In order to allow for appropriate control measures, sensible techniques are required to effectively determine UCs.

This relation is established by tests incorporating questionnaires or to find out about users' opinions on certain aspects to the media's qualities presented. The quantitative result of such an assessment is called a *Mean Opinion Score* (MOS), which is usually obtained by *subjectively rating* stimuli with respect to a criterion like inter- or intra-media qualities in a presentation. Subjects express their judgements of media qualities according to a give scale. Finally, the scores are averaged across subjects to obtain the final MOS [11].

2.3.3 Assessment of subjective QoS

When dealing with data networks together with interactive applications using them, a distinction between objective and subjective QoS must be made. *Quality of Service* usually denotes properties of the network that can be *measured* by running experiments and observing the behavior of the network traffic and the application behavior. In order to derive abstract estimates like *high* or *low quality*, the measurements must be related to the context of the used applications.

However, the measured QoS metrics primarily denote *objective* metrics, i.e., they are related to the measured items, for instance protocol PDUs, bytes, video frames etc. On the other hand, a human observer does not think in terms of frames per second, throughput etc., but rather observes the used application and then derives his own *subjective* QoS measure for it, taking into consideration the audio/visual and logical output of the application.

For instance, a video framerate of 25 frames per second (fps) normally would be considered as high quality. However, if the video is highly compressed, then compression artefacts will be visible, for instance compression blocks or mosquito noise, and the human observer would surely rate the presented video to have a *low* quality.

Within workpackage JRA.6.1, in addition to objective QoS, we want to focus on subjective QoS as rated by human observers. The main goal within this research area is to find

mappings from objective QoS metrics to subjective QoS. When measuring subjective QoS, different scales can be used. On a continuous scale, usually the intervall [0,100] used, 0 denoting the worst, and 100 denoting the best subjective QoS. For discrete scales, for example five-point scales (*excellent*, *good*, *fair*, *poor*, *bad*), 9-point scales (5-point scale plus 4 points in-between two points) or 11-point (9-point scale plus one point above execllent and one below bad) can be used [12]. In [13] also a 7-point scale for relative comparisons of two different videos is described.

In principle two different methods for deriving mappings between objective and subjective QoS can be used. First, a large number of observers is asked for their opinion, for instance by letting them rate a certain video on a scale between 0 and 100. Computing the *mean* of all ratings results then in the MOS, which denotes a hopefully meaningful estimate on how human obeservers on the mean rate the observed QoS. Unfortunately, this approach suffers from several drawbacks. First, human observers may drastically differ in their rating, either due to different perception, different abilities to focus on the experimental task, different audio/visual abilities, differing tastes for music etc. This results in a rather high variability of subjective judgements. Thus, a large number of experiments is necessary in order to derive stable estimates with small confidence intervals. Second, some subjective ratings must be considered as outliers due to inconsistent ratings, which for instance is the case if a person rates a low-bitrate video with visible compression artefacts much better than a high-bitrate version of the same video without any artefacts. Care must be taken in order to identify and remove such outliers without endangering the overall estimate. Thirdly, often relative trends in subjective ratings are consistent, but the absolute numbers differ significantly. Again, care must be taken to rescale subjective ratings to one single niveau without endangering the meaning of the MOS.

The second princpal method for finding mappings from objective to subjective QoS is to use a small number of experts, or even only one expert. Of course such an experiment would only represent the judgement of one single individual or a small number of individuals, and it is questionable whether these results represent the mean judgement of human observers accurately. However there are indications that such expert based experiments not necessarily yield bad results.

Shortcomings of MOS are identified in [11]. As an alternative, *Task oriented Performance Measures* (TPM) are proposed. Here, the subjects are exposed to different levels of the stimuli (e.g. different frame rates), and the outcomes are measured objectively. The performed task is related to a given context and the measured performance is thus relevant to an application that requires this task. This represents an operationalized direct way of dealing with the subjects' percepts such that the additional level of self-reflection is removed and validation of the obtained data is alleviated.

A project dealing with *subjective quality assessment* with ratings of video performed by real users is presented in Section 3.4. Section 3.5 sketches a framework for *pseudo-subjective assessment*. In principle, users are simulated by a *Random Neural Network* (RNN) that is trained to reproduce the relation between the parameters affecting the quality and the perceived quality itself. Thus, this method represents a hybrid approach combining subjective and objective rating.

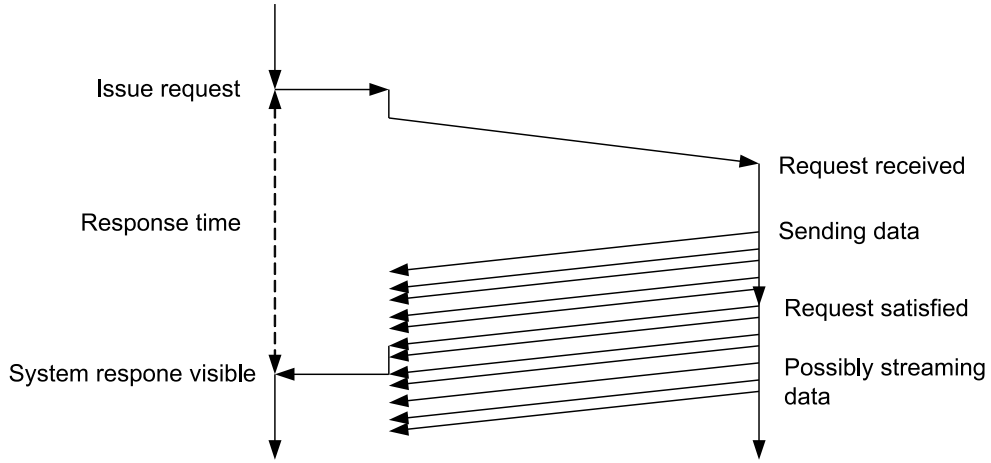


Figure 2.3: Definition of response time.

2.3.4 Subjective response time QoS

Figure 2.3 shows the definition of response time, being the time between issuing a request to the system until the result is visible (or audible) to the user.

Response time is influenced by the time it takes to transfer the request to the remote server, the time the remote server needs for satisfying the request, and the time it takes to transfer and present the request to the end user. In [14] three important limits for response time are given:

- 0.1 second is about the limit for having the user feel that the system is reacting instantaneously, meaning that no special feedback is necessary except to display the result.
- 1.0 second is about the limit for the user's flow of thought to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data.
- 10 seconds is about the limit for keeping the user's attention focused on the dialogue. For longer delays, users will want to perform other tasks while waiting for the computer to finish, so they should be given feedback indicating when the computer expects to be done. Feedback during the delay is especially important if the response time is likely to be highly variable, since users will then not know what to expect.

From intuition it is clear that longer response times decrease user satisfaction. It is, however, generally not easy to quantify the user satisfaction as a function of response time. For instance, given a scale from 0 to 100, 0 denoting a dissatisfied user, and 100 total satisfaction, on average how would a response time of 5 seconds be rated? In order to be able to quantify the user satisfaction as a function of the response time, results from the scientific literature can be used. In [15] the average attention span window is defined

Table 2.4: Exit rates depending on latency.

< 7 seconds	7%
8 seconds	30%
> 12 seconds	70%

Table 2.5: User satisfaction depending on response time.

Rating	Scenario 1	Scenario 3
High	0–5 seconds	0–39 seconds
Average	> 5 seconds	> 39 seconds
Low	> 11 seconds	> 56 seconds

to last for 4 seconds. Web downloads lasting longer than 4 seconds are said to bore users. The authors however do not justify this definition. The same rule is given in [16], citing Forrester and Information Week, June 5, 2000. In [17], a premium class of Web users is defined requiring download times to be less than 5 seconds.

The most popular Web response time rule has been reported by [18], setting 8 seconds as the limit users are willing to wait for Web downloads. Zona Research has extended this 8 second rule later to a mapping of latency to expected exit rates (Table 2.4).

Zona also states that 20 % of users exiting are lost and will not come revisit the Web site. This is an important fact that can be included into the construction of business cases. Finally, in [19], the minimum requirement for Web downloads is a latency < 11 seconds.

More advanced research states that the user perceived QoS is not only a function of the response time, but also depends on the user's expectations [10]. In [2], Web response times have been rated for different scenarios using a scale low, medium, and high. In Scenario 1, no progress of current downloads was visible. In Scenario 3, downloads were incremental, and downloaded Web page components were immediately visible (Table 2.5).

A more general subjective rating by 30 individuals of latencies is shown in Figure 2.4. It can be seen that the low-rating coincides with several results from other studies. In further studies it was stated that the maximum tolerable latency is not fixed but depends on factors like the length of the ongoing session [20]. This tolerance will drop slightly as time advances.

Such thresholds are found may serve as parameters for Service Level Agreements. As pointed out before, the goal is to provide technical parameters that mirror user perception of quality. Such parameters are important for both service providers and users, as they reveal the degree of conformance between promised and real quality. However, most users might have problems in understanding parameters such as delay quantiles or loss ratios, and they might not either be interested in such technical facts. In case it should be necessary to distinguish between application and network performance (e.g. in case of different providers), and given the application does not report problems in an explicit

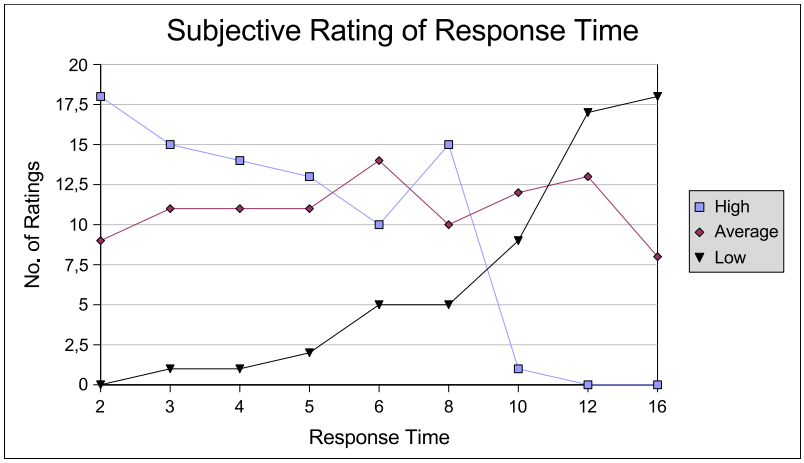


Figure 2.4: Subjective rating of response time.

way, users need somewhat intuitive tools and indicators to tell them about type and severity of potential problems mainly on network level (see Section 3.6). It is also worth noting that such tools and indicators could help applications to monitor and manage the QoS. If one would be able to exclude network malfunctioning (that is in general only to be observed indirectly through the application), the application would be left to be blamed. Another possibility is to improve the network support for QoS (see Section 3.2).

2.3.5 Utility functions and bandwidth auctions

In order for the rational players of a game – or an auction – to get what they really want, they need a way to express their relative preferences for the various outcomes of the game. To this end, an appropriate mathematical tool is used; namely the *utility function*. This is a function that reflects the ordering of user preferences regarding the various outcomes of the game by assigning to each outcome a value. For example, the utility function $u(x)$ of a customer who wishes to purchase bandwidth, defines the customer preferences for acquiring various quantities x of bandwidth. It is henceforth assumed that it is associated with the customer's willingness to pay for the respective quantity of bandwidth. Certain typical utility functions are:

- *Guaranteed*, pertaining to customers demanding a specific quantity of bandwidth, q_g;
- *Linear*, pertaining to customers that are satisfied with any quantity of bandwidth up to a maximum q_{max} and can only afford prices below a certain threshold, which equals the respective slope of their utility;

- *Elastic*, i.e. pertaining to customers with a concave utility function representing diminishing return as the quantity of bandwidth increases. Thus, elastic customers purchase various quantities of bandwidth but each additional unit is of less value to them compared to that attained by the previous unit.

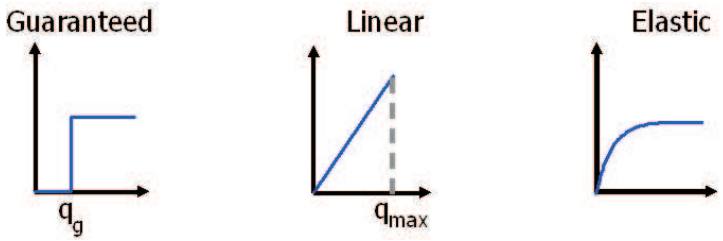


Figure 2.5: Users' utility functions

When a customer decides to purchase a quantity of bandwidth x, the amount to be paid for that quantity, namely the cost $c(x)$ is also to be taken under consideration. The difference of the utility minus the cost is defined as Net Benefit, thus $NetBenefit(x) = u(x) - c(x)$. Maximization of Net Benefit is often assumed to be the objective of a player participating in a game such as an auction or a negotiation.

On the network level, a dedicated bandwidth that is provided on a semi-permanent basis might eliminate some of the risks associated with the statistical bandwidth sharing in best-effort Internet. Equipped with CAC, the user would get reliable network service in terms of delay, loss and goodput in case the fixed-bandwidth connection could be established – this situation is well-known from classical telephony. However, besides the fact that resource reservation on an individual basis is more or less impossible in best-effort Internet if one does not consider extensions such as MPLS, a "circuit-switched"-type data network usually does not allow for high loads due to the absence of statistical multiplexing gain, which makes that bandwidth rather expensive for the customer. One way out of this problem consists in auctioning bandwidth on-line (see Section 3.7), which combines reservation features with statistical multiplexing gain.

2.4 QoS management solutions

QoS management is an important issue for users, service and network providers. In case an SLA has been established, all the parties need to know whether the service behaves as expected with regards to speed, accuracy and reliability [1]. But even in best-effort scenarios with no explicit SLA, there are certain minimal requirements that have to be met so that a customer "perceives" connectivity at all. This implies that the (perceived) quality has to be monitored and fed back to the different partners in order to make the quality control loop work efficiently.

Best-effort Internet has another implication – in case of resource shortage, applications tend to time out, making unconscious users retry and worsen the situation even more by undeliberately carrying out Denial of Service attacks. Catastrophies such as September 11

usually lead to break-down of services and networks – many people re-try, but virtually no one is getting any service anymore. Especially in the context of e-Business, drop-outs might cause severe damage in the trustworthyness of such a system, simply because people's money is involved. Signalling overload problems to users might cause them to be patient, relax and to retain trust into the system: "I think it's great...saying we are unusually busy, there may be some delays, you might want to visit later. You've told me now. If I decide to go ahead, that's my choice." [2].

In the following, we review the state-of-the-art regarding quality management and feed-back in the Internet. Such a feed-back is mostly related to problems and abnormalities; in general, the partners "keep quiet" if everything behaves as expected. We assume the cut between application and network between OSI layer 7 (application-oriented protocols) and 4 (transport protocols), respectively. The notion "network" may comprise several IP networks belonging to different ISPs/IBPs. Figure 2.6 illustrates feed-backs that are discussed in the following; the numbering matches that of the items.-

1. Feedback from the network (i.e. OSI layers 1–4):

    a. Network → application: The network makes the application suffer from problems (implicit feedback), as packets are delayed or lost. In general, no explicit feedback about such problems is provided by the network (e.g. by sending signalling packets). However, applications have the possibility to measure the performance and adapt themselves to the conditions within the networks (see 2.1).

    b. Network → network provider: This is usually done through load monitoring by SNMP on rather long time scales (several minutes) by polling devices or receiving traps. If the aggregate load on a specific link exceeds a certain, mostly experience-based threshold quite frequently, that link's capacity is upgraded, i.e. "bandwidth is thrown onto the problem". Generally, a network provider (ISP or IBP) just cares about the own network and monitors the links towards other providers as if they were local.

    c. Network → user: The user feels network problems in an implicit way through the application, but is seldomly informed directly e.g. through warnings or error messages issues by the operating system (such as "cable disconnected"). There are rather rudimentary tools such as `ping`, `bing`, `pathchar` or `traceroute` available in most operating systems, in some cases even a bandwidth monitor. However, the information presented is rather cryptic and needs expert knowledge to be interpreted. Users would more likely need indicators telling them whether the network status matches the SLA or not, see Section 3.1. Section 3.6 proposes a performance indicator that vizualizes the impact of the network on the bit rate perceived by a connection.

2. Feedback from the application (including OSI layers 5–7):

    a. Application → application: Implicit feed-back is given in a way that the interaction of processes belonging to a distributed application is influenced by the interconnecting network(s). As pointed out before, some applications measure

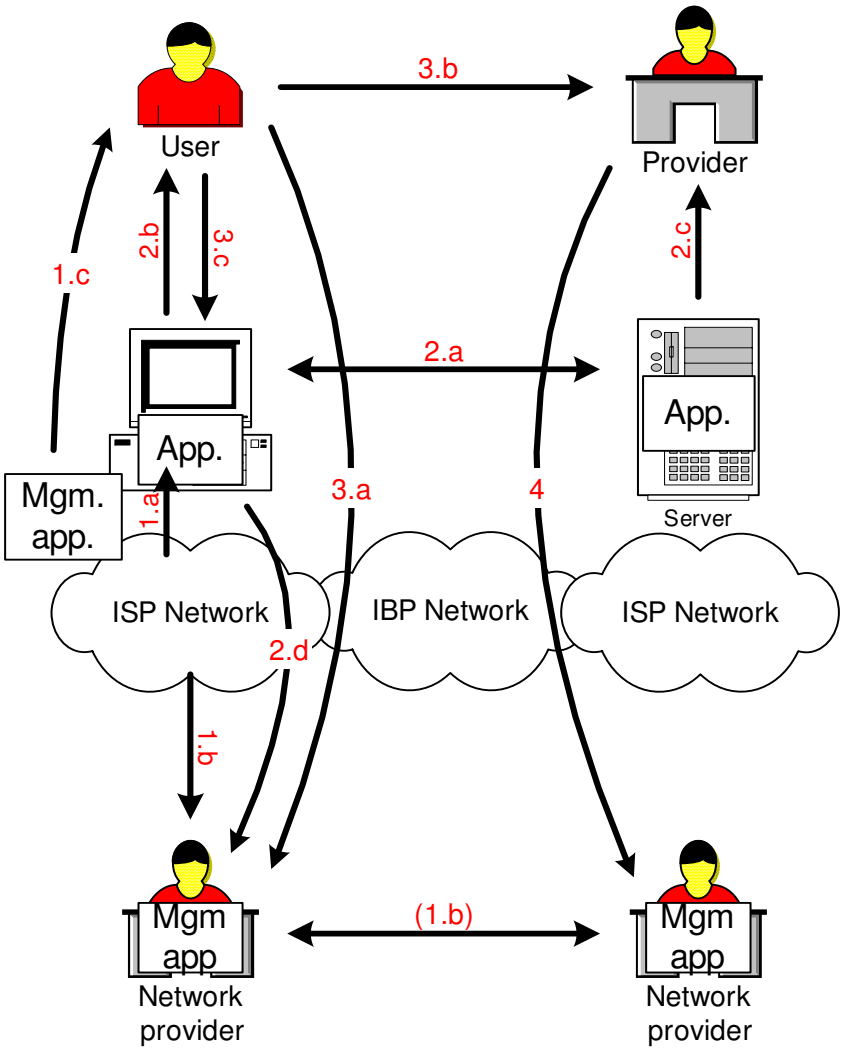


Figure 2.6: Overview of quality feed-backs; for numbering of the arrows, please see text.

the network impact. For instance, the application protocol RTP [21] allows for including sender and receiver reports that are evaluated e.g. by a video-conferencing application [22] in order to adapt the coding to the network conditions. Section 3.8 proposes a method for adapting the play-out buffer for Voice over IP based on predictions. Yet another kind of feed-back are warning messages such as web server overload that might reflect both application and network problems.

   b. Application → user: The implicit feed-back is a consequence of 1.a and 2.a, respectively. The user might get to feel the applications-perceived problems as far as the application is not able to compensate for problems originating from the network. On the other hand, the user might experience application-related problems while the network is healthy. In any case, icons such as "hourglass" and progress bar or warnings might be displayed, e.g. by the video-conference application or the web browser.

   c. Application (server side) → service provider: The functioning of a service is observed e.g. through issuing test requests. However, this does not necessarily reflect the quality in terms of speed, accuracy and reliability that is perceived by the customer.

   d. Application → network provider: The network provider might sniff for special packets containing application-level status information (such as RTCP send and receive reports).

3. Feedback from the user:

   a. User → network provider: A typical user reaction consists in blaming the closest network provider (ISP) for any kind of trouble with the networked application that is experienced. Especially if the user's connectivity is affected, this reporting has to be done by other means of communication, e.g. by phone. However, in best-effort Internet, the situation can be quite complex. There may be several providers that have to be addressed and that use to be convinced that the problem is not to be found in their part of the network. Given their quite limited possibilities of monitoring (cf. 1.c), an average users might find it hard to find out about the real nature of a problem.

   b. User → service provider: On some web sites, users are welcomed to leave comments about the content. On [23], users are asked about their connection speed in order to adapt the web pages to their facilities. However, there seems to be a trend that users inform providers about problems rather implicitly (by giving up using a service) than by providing explicit feedback.

   c. User → application: Some applications allow for explicitly changing settings in order to cope with problems, e.g. by lowering the bit rate of a video conference in order to make the stream more robust to jitter. Again, the implicit way of dealing with the problems is to give up using the service.

4. Feedback service provider → network provider: Upon perception of quality problems and/or user complaints, a service provider might contact the correponding network provider in order to ensure the quality of the service's network connectivity.

Chapter 3

Selected Contributions

This chapter contains a selection of results and views on the topic of this deliverable as contributed by partners of the Euro-NGI WP.JRA.6.1. This material shows the breadth of expertise among the partners with regards to the topic of interest and is intended to serve as a basis for further joint research work.

3.1 Telenor Activities

Terje Jensen
Telenor, Norway

A number of activities are in some sense related to the scope of WP.JRA.6.1; addressing key issues such as QoS parameters, service level requirements, performance assessment, Service Level Agreements and functionality in order to configure resources and estimate performance. A few of these are elaborated in the following. Note that they are all related as illustrated in Figure 3.1.

3.1.1 QoS, service requirements

In order to provide and configure the network resources it is vital for a network operator to assess characteristics of services to be provided. This also includes QoS requirements of services. In particular this goes on any IP-based service, although some emphasis is also placed on services delivered by wireless access – being mobile or WLAN. Besides used as input when designing systems, guidelines on conditions to place in Service Level Agreements are obtained. Here, these conditions are mostly related to technical aspects, as other aspects as well have to be considered when setting up an actual SLA.

Some support for estimating service characteristics is found in standardisation documents, e.g. 3GPP, in addition to other published papers. A main challenge seems not finding relevant material, but rather to present the requirements in a systematic manner. Typical QoS requirements can be divided into

1. delay-related
2. loss-related

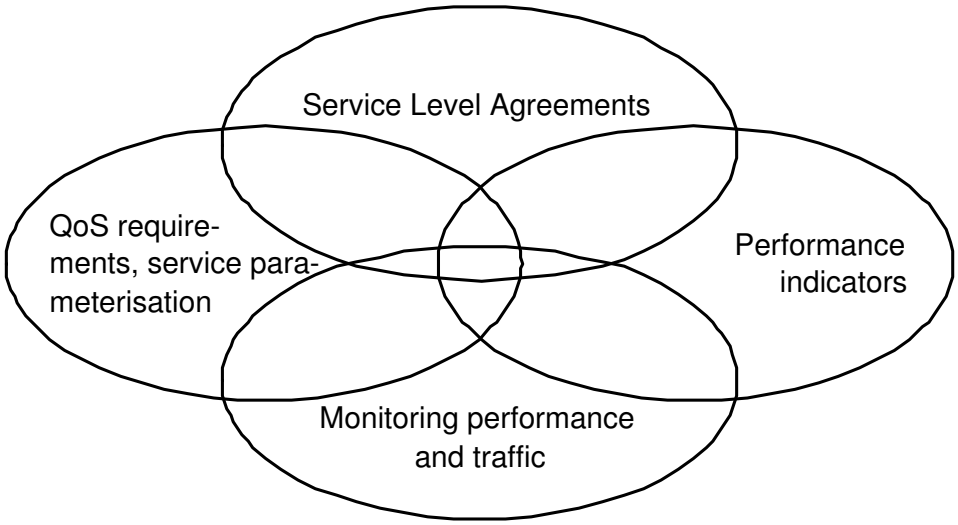


Figure 3.1: Illustration of selected activities

3. dependability-related

All these have to be considered, although the third area is less frequently covered in standardisation documents.

As a basic approach to the QoS topic some fundamental results have been elaborated jointly with other European operators (ref. EURESCOM P806 project [5]). The results have been published at different conferences and also provided one of the main fundaments for ITU-T Recommendation E.860 [1]. The scope and motivation for that work was to solve the "generic QoS understanding" in a multi-provider environment also considering the multi-service and multi-technology setting. Hence, a rather fundamental and generalized interpretation of QoS was needed. In fact the definition chosen – QoS = degree of conformance of the service delivered to a user by a provider, with an agreement between them – brings the quality understanding and management of internet/telecommunication in alignment with other industries. It also straightens the confusion between service levels/service classes and QoS. Working in a commercial environment it is important to arrive at clear interpretation of such essential terms as QoS.

Another important element in describing service characteristics is defining components of services. At a higher level, a service that a user faces would likely be composed of a number of components – each with its specific characteristics. Addressing this area in an efficient manner, a framework for composing services is asked for. Several proposals can be identified in publications, in particular from international fora, although commonly restricted to certain aspects of the service provision. The full-blown provider situation has to cover all aspects from advertising and marketing to operation and customer complaint handling. However, one mostly focuses on the network- and operational-related aspects.

An example of composition is a multimedia session that could well be composed of a video component, an audio component and a number of data components. Again, each of these components could have different characteristics. An end-user would frequently relate to the composite behaviour of the components, which make up the complete service. Moreover, linkage between service and application of the service must be considered. Here, a distinction is made according to the understanding that service is something that is "exchanged" between entities (typically a user and a provider). Therefore, in principle, a service could be charged for. Application, on the other hand, is a unit making use of the service. An example is the service called 64 kbit/s circuit switched connection. This can be applied for voice, fax, modem, etc.

Parameterisation of service components is then possible, both considering the usage situations as well as how the services are implemented. In some cases no strict bounds are given for services, hence allowing flexibility in the service delivery. An example is the throughput provided for a TCP session. For dimensioning purposes the application/usage of such services must be considered, that is taking into account that some minimum service levels are commonly expected. Again, this is done for several access types – wireless and wired.

3.1.2 Performance indicators

Managing any network or service provision, defining and following an adequate set of performance indicators is a necessity. Such indicators are typically used in order to assess the "health condition" of the operation and service delivery. In general both technical and financial indicators will be used, as well as others, e.g. reputation and so forth. However, technical-related ones are the main topic here. Again, these could be divided into separate parts, for example referring to different portions of a system and different phases of the service provision.

A main challenge of performance indicators worked on is to devise a set of indicators reflecting the service levels as experienced by the users. Initiating an activity on these topics, it seems like a framework for handling performance indicators was missing. Hence, elaborating initial ideas for such a framework was part of the first steps to take. Naturally assessing the indicators, monitoring is a pivotal part. A number of measurement installations would likely be installed in most operations in order to follow performance of different areas. How monitoring apparatus can be efficiently combined is therefore one of the key questions. Again, a result should be reflecting the end-user experience.

A basic idea allowing for a swift arrangement is to re-use monitoring equipment and observations for different objectives, one objective being to follow performance indicators. This, however, places a further challenge on the performance indicator collection, as a number of under-lying parameters might need to be aggregated in order to estimate an indicator value. Having (almost) independent observations for different portions, the end-to-end view observation may not be trivial to estimate. Therefore, some effort has to be placed on those matters.

The main systems looked at are 2.5/3G mobile, i.e. GSM family and UMTS. In addition to the challenges found in wired access systems, varying radio conditions may also severely impact the end-user experience. These would likely differ in time and geography as well as be influenced by the load in the system (that is presence of mobile users).

A further prioritisation of services may reveal that non-voice/non-video services should be examined firstly. An argument for this is that voice and video have been evaluated for some time and technical parameters' influencing the user experiences tried to be assessed. Fewer results seem to exist on other service types – which in the mobile context are SMS, MMS, WAP, download, etc.

Looking at the implementation of several of these services, different system portions can be identified. Moreover, it is also seen that some services can utilise others – for example MMS may apply WAP-push in order to deliver the message to the receiving mobile user. As mentioned earlier, a basic question is whether following performance for the different portions allows for estimating the performance of the "more aggregate" service. This motivates for looking at several basic statistical issues for collecting and aggregating samples.

One of the portions of a 2.5/3G system is the packet-based core network. On the longer run an "all-IP" network is also foreseen. Therefore, most topics addressed would also be relevant for service provision on wired access. In fact, is could well be a working hypothesis

that the wired access compose a subset of the area looked into.

Another fundamental question is how to present the performance indicator values. Keeping in mind that there are several types of receivers of the indicator observations, different presentation forms could apply. For example, a technician would likely want of see absolute observations in order to decide whether or not any failures have occurred. On the top management, however, more relative values could be presented. This could be obtained, for example, by relating an observation to a target value. That is, a observation could relate to a reference "100 points", where anything above is better than target and anything below is worse than target. More thresholds could also be defined in to decide on other actions.

3.1.3 SLA template and conditions

A steadily increasing awareness among customers is observed regarding conditions in Service Level Agreements. This refers both to residential and enterprise customers. In order to alleviate the process of defining SLAs, an appropriate structure and template should be defined. A start on this was undertaken by EURESCOM project P806 proposing a structure of the QoS-part of an SLA. The following main items are included in that part:

- Service description including the interface at where the service is delivered
- Quality of Service parameters and values
- Traffic conditions – or service usage conditions during which the QoS is to be obeyed
- Measurement arrangements for monitoring QoS and traffic conditions
- Reaction patterns describing actions to undertake in case any of the conditions are broken (examples being discount, traffic throttling, etc.)

This structure, together with samples of applications are described in P806 deliverables. Although some time has passed since then, it seems like the ideas are gradually emerging in different bodies, such as ITU-T Rec. E.860 and a joint project between Norwegian telecom users association and the Norwegian regulatory authority.

As mentioned earlier, a clear definition of QoS is necessary for this work. Later results have been successful relating this understanding with other concepts such as applying the eTOM reference model[1].

Although addressed by several EU projects (including Tequila, Aquila, Cadenus), a basic framework for SLA does not seem to be coherently described considering IP-based services. This refers to the complete end-to-end story both addressing individual customers and inter-provider aspects. In particular the IP-based service provision configurations (on both wired and wireless access) allows for several additional challenges not previously seen for other systems. One aspect is to include SLA in the eCommerce activities (B2B, B2C, C2C) to the extent feasible.

[1] www.tmforum.org

3.1.4 Functionality in nodes and devices for "verifying" performance levels

Monitoring traffic flows and service levels has been an activity for quite a few decades. Still there seems to be strive for finding the proper balance between achieving an adequate picture of conditions in the system and not spending too much resources on monitoring. One centralised approach is to monitoring servers and common network resources. This may save some monitoring equipment, although too many averaging operations might hide problematic portions. A fully distributed approach is to have monitoring agents installed in user devices, although then a management challenge would be seen together with the "trust level" between the user and the provider.

Considering the multi-service, multi-technology, multi-provider situation seen by a Next Generation Internet, the monitoring challenge will grow further. A systematic analysis of the different monitoring options could be undertaken to provide basis for selecting the ones for realise. In particular it is seen that different monitoring arrangements would likely be the better ones depending on the different phases of service provision - for example the arrangements for a "mature" service might differ for arrangements during initial roll-out.

A specific objective is to apply the monitoring results to trigger certain actions, either by the operator/provider or by the user. Multiple purposes could be defined, both on enhancing the capacity (or re-configuring the available capacity) or restricting the traffic load (admission control, policing, charging, etc.).

3.2 Network Support for QoS for IP-based Applications

Hermann de Meer
University of Passau, Germany

Users wish to have access to a whole plethora of telecommunication and data communication services via the Internet; they wish to access an Integrated Services Network (ISN). However, the Internet and IP was never designed to handle such traffic and so the Internet community must evolve the network and enhance the Internet protocols in order to cater for the needs of these new and demanding applications. In this section, we try to understand about QoS for IP-based applications and how the network must be changed to support these new applications.

To provide support for real-time applications, we need to introduce mechanisms at many different parts of the communication stack. At the network layer, we need to modify router behaviour so that packets belonging to QoS sensitive flows receive some kind of preferential treatment, compared to "normal" data packets. We also need to modify the behaviour of routing protocols in order to support multicast communication and QoS-based routing metrics. At the transport layer, recall that we only have two general protocols: TCP for traditional applications that require an ordered by-stream delivery, and UDP for applications that build in specific control mechanisms at the application layer. For real-time flows, we can identify some general requirements, which we will see can be implemented by extending UDP as in the Real-time Transport Protocol (RTP). At the application layer, we may identify other mechanisms that are required for specific real-time applications: floor control for conference applications; transcoding for audio and video flows; security mechanisms such as authentication. Although it is possible to identify some general requirements, such higher-layer mechanisms tend to specific to particular applications. Here, we consider the support that we have in the network and at the transport layer, as well as some general issues concerning the interface between the application and the network. Why do we not consider the link layer and physical layer? Surely these have a fairly vital role in QoS as they provide the transmission capability? Remember that IP tries to hide the lower layers, so although we will see there are important issues concerning the lower layers, we concentrate on the network layer and transport layer.

Even if we could offer some sort of QoS control mechanism, with prioritisation or traffic differentiation, there is then the issue of pricing. How do we charge for use of network resources for a particular treatment of traffic for a particular customer?

So we can ask ourselves several questions. Firstly, can we provide a better service that that which IP currently provides – the so-called best-effort? The answer to this is actually, "yes", but we need to find out what it is we really want to provide! We have to establish which parameters of a real-time packet flow are important and how we might control them. Once we have established our requirements, we must look at new mechanisms to provide support for these needs in the network itself. We are essentially trying to establish alternatives of FCFS for providing better control of packet handling in the network as well as trying to support multi-party (many-to-many) communication. We also need to consider how the applications gain access to such mechanisms, so we must consider any

application-level interface issues, e.g. is there any interaction between the application and the network and if so, how will this be achieved. In all our considerations, one of the key points is that of scalability – how would our proposals affect (and be affected by) use on a global scale across the Internet as a whole.

The Internet was never designed to cope with such a sophisticated demand for services [8]. Today's Internet is built upon many different underlying network technologies, of different age, capability and complexity. Most of these technologies are unable to cope with such QoS demands. Also, the Internet protocols themselves are not designed to support the wide range of QoS profiles required by the huge plethora of current (and future) applications. In [24], the authors speak of the Internet evolving to an integrated services packet network (ISPN), and identify four key components for an Integrated Services architecture for the Internet:

1. service-level: the nature of the commitment made, e.g. the INTSERV WG has defined guaranteed and controlled-load service-levels (these are discussed later) and a set of control parameters to describe traffic patterns, which we examine later;

2. service interface: a set of parameters passed between the application and the network in order to invoke a particular QoS service-level, i.e. some sort of signalling protocol plus a set of parameter definitions;

3. admission control: for establishing whether or not a service commitment can be honoured before allowing the flow to proceed;

4. scheduling mechanisms within the network: the network must be able to handle packets in accordance with the QoS service requested.

A key component that is required here is signalling – talking to the network. Signalling is essential in connection-oriented networks (used for connection control), but datagram network typically need no signalling. No signalling mechanism exists in the IP world – it is not possible to talk to the network, one simply uses the service it provides. The signalling part of a connection-oriented network communication offers a natural point at which information about the particular requirements of a connection can be transmitted to the network. As IP is connectionless, any signalling mechanism should ensure that it is compatible with current operation of the Internet and should not constrain or change the operation of existing applications and services.

The simple description of the interactions between these components is as follows:

- A service-level is defined (e.g. within an administrative domain or, with global scope, by the Internet community). The definition of the service-level includes all the service semantics; descriptions of how packets should be treated within the network, how the application should inject traffic into the network as well as how the service should be policed. Knowledge of the service semantics must be available within routers and within applications.

- An application makes a request for service invocation using the service interface and a signalling protocol. The invocation information includes specific information

about the traffic characteristics required for the flow, e.g. data rate. The network will indicate if the service invocation was successful or not, and may also inform the application if there is a service violation, either by the application's use of the service, or if there is a network failure.

- Before the service invocation can succeed, the network must determine if it has enough resources to accept the service invocation. This is the job of admission control that uses the information in the service invocation, plus knowledge about the other service requests it is currently supporting, and determines if it can accept the new request. The admission control function will also be responsible for policing the use of the service, making sure that applications do not use more resources than they have requested. This will typically be implemented within the routers.

- Once a service invocation has been accepted, the network must employ mechanisms that ensure that the packets within the flow receive the service that has been requested for that flow. This requires the use of scheduling mechanisms and queue management for flows within the routers.

Imagine a video application. The application or user would select a service level and note the traffic characteristics required for the video flow. Information such as required data rate would be encapsulated in a data structure (service interface) that is passed to the network (signalling). The network would make an assessment of the request made by the application and consider if the requirements of the flow can be met (admission control). If they can be met routers would ensure that the flow receives the correct handling in the network (scheduling and queue management).

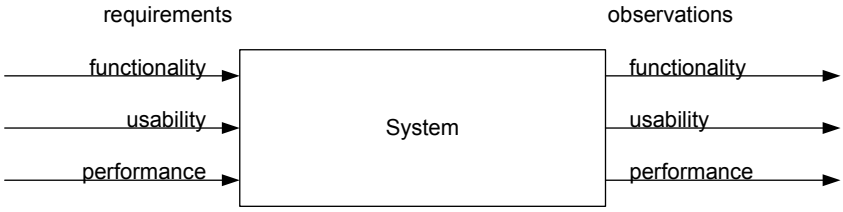


Figure 3.2: Different types of user expectations.

3.3 Linking Quality of Service and Usability

Gabriele Kotsis and Thomas Grill
Johannes Kepler University, Linz, Austria

3.3.1 Motivation

A user who is accessing a system or is using a service approaches specific tasks and the associated media for their fulfillment with well defined expectations to the system. The expectations of the user can be classified with respect to three different aspects, namely regarding the functionality of the system, regarding usability and ease of interaction, and regarding performance of the system (see Figure 3.2).

When designing and implementing a system, those expectations need to be considered in the specification of qualitative and quantitative requirements as well as in evaluating the system with respect to the expectations and requirements. Most work on quality of service is mainly focused on the last system aspect and tries to map and provide system performance to meet user requirements and service levels. Recently, more emphasis was given to user perceived QoS but mainly focussing on translating system level performance metrics and concepts to the user level. Here, we will investigate approaches for user perceived QoS management but will focus on the interdependencies between usability/ease of interaction and performance/QoS.

We will discuss metrics of user perceived QoS which are needed for identifying user expectations and requirements. We will argue for approaches that also consider context information in defining QoS level agreements and that user perceived QoS must not only take the technical aspects of QoS requirements into account, but must also consider usability aspects. With respect to QoS provisioning, we will present a user oriented proactive approach for QoS management.

3.3.2 What does the user "perceive" as QoS?

One of the most important parameters that defines the quality of service for the user are the users subjective expectations. Her expectations may be based on the interest on the

task and the expected quality the user perceives may also vary with the importance of the task. Ongoing work deals with identifying service level agreements and metrics for defining the interest of a task in coherence with the users expectations [25].

Another approach is to define user perceived QoS in terms of monetary units, linking cost models and user perceived QoS. Depending on the expected user perceived QoS an applicable pricing model (see Deliverable D.JRA.6.2.1) may be defined. The pricing model influences the importance of a service for the user while his expected user perceived QoS could be defined as follows.

$$\text{User satisfaction} = f(\text{applied pricing model}, \text{expected QoS}, \text{user perceived QoS})$$

According to the formula given above the "user satisfaction" is an indicator if a user would use an offered service for a specified price. If the user satisfaction applied to a market model is too low, there is no or not enough market for the service and thus no need to think about implementation and QoS parameters for the service. Analysing the gap between expected QoS and user perceived QoS will serve on the other hand as an indicator for the price a user is willing to pay. If there is a gap between expected QoS and user perceived QoS, the service might still be satisfying assuming that it is offered for a very low price or even free of charge.

The models described so far can be described as static models in that they are basically not able to represent changes in user expectations. But the requirements and expectations are typically not static and also exhibit interdependencies. Therefore, such systems would require a continuous monitoring, evaluation, and possibly adaptation. Research in ambient intelligence or pervasive computing is investigating those issues mainly focusing on the adaptation of functionality and interaction introducing the notion of context awareness. This concept is also of interest for QoS modelling and provisioning [26].

The user accesses the services offered by the system within a specified context that influences the parameters of the QoS expectations for the user. Extending the notion of user satisfaction as given above with a model for context awareness, requires a context-based representation of expected QoS. We therfore suggest to define the expected QoS not only as a function of the QoS parameters under consideration (e.g. a certain threshold for delivered frames per second in a video conference application) but also indexed with context information:

$$\text{expected QoS} = f(\text{QoS parameters}, \text{context})$$

Specifically, web [27] and multimedia [28] [29] applications have been studied in order to identify (user-oriented) QoS parameters.

Typical examples for context are time or location, but may also include the application domain (e.g. financial transactions versus gaming applications), maybe the age of the user and many other factors (the interested reader is refered to the discussion of context in the pervasive and ubiquitous computing community).

User perceived QoS depends on the workload of the system and on the performance characteristics of the system but is strongly influenced by the way the system presents itself to the user, which is commonly refereed to as usability of the system. We therefore

define user perceived QoS as follows:

$$\text{user perceived QoS} = f(\text{load}, \text{system}, \text{usability})$$

Usability itself doesn't deal in any way with the assurance of enough resources. The aspect that is important and necessary in respect to usability terms is how the user perceives a certain functionality that is defined in the appropriate interfaces and transported to the user via a user interface.

User perceived QoS and usability are thus very much correlated [30]. A user may experience problems in interacting with the system caused by either a poor performance of the system or by deficiencies in the interaction design. For the user, the actual cause is not always transparent resulting in a general dissatisfaction with the system. For system providers it is therefore difficult to improve the system if the deficiencies are not clearly identified. We argue that a better understanding of the interdependencies between usability and QoS will help in identifying deficiencies.

3.3.3 Proactive user oriented QoS provisioning

QoS management requires on line mechanisms for observing and adapting the system behaviour in order to meet QoS requirements. State of the art QoS research therefore targets primarily the provision of sufficient resources for specific task requirements. These efforts consist in measuring physical parameters, resource utilisation and in finding and applying means and methods to provide sufficient resources as e.g. bandwidth. One of the challenges is in providing end-to end Quality of service (see e.g. [31] for mobile networks or [32] studying IPv6 networks).

In this work, we are also considering end to end QoS but are proposing a distributed, proactive approach [33, 34] that tries to predict future system states based on local observations on the component under study. The suggested generic architecture is depicted in Figure 3.3.

Each component contributing to the end-to-end QoS chain may be enhanced by a performance management component.

Sensors collect context information in order to obtain and process knowledge about the environment, e.g. the type of task currently being performed by the user. The users context is to be used as a pool of input data that could be applied to the QoS aspects in a way to proactively regulate the requirements of the user respectively a service by means of the users context and context-changes. Sensors are also used to collect and transmit information about user expectations and information about the current (performance) state of the system.

Based on this observations in the past, a forecasting component tries to identify future states of the system. This forecasting component can apply simple statistics such as a moving average on performance data, but may also be implemented as an intelligent agent trying to study and predict user behaviour. Again, techniques and methods from other research disciplines such as ambient intelligence (AmI) are worth to be studied (see for example [35] for an overview of agent technology in AmI).

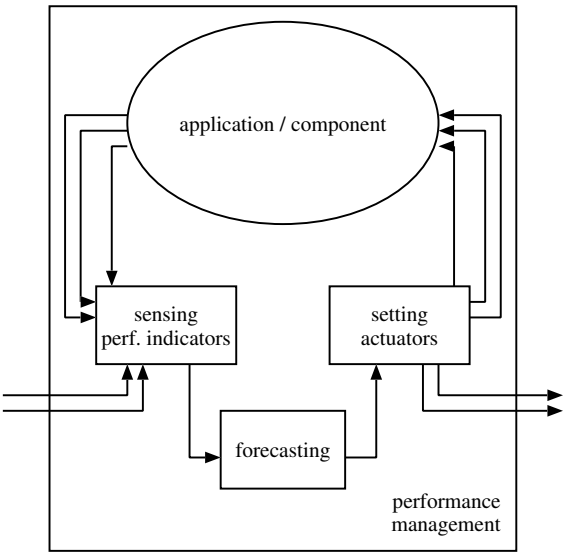


Figure 3.3: Proactive Performance Management.

Finally, actuators will take appropriate actions in order to improve and control the performance and behaviour of the system in order to ensure that the user expectations are met. In this phase, feedback to the user is an important issue and again usability concerns come into play and need to be studied when trying to identify appropriate feedback mechanisms. An example for QoS feedback to the user may be an indicator that indicates the buffer state of a video streaming application. The user will be satisfied as long as the bufferring is faster than the played video stream. Additional feedback will be provided to him. This puts power into the users hands as it enables the user to evaluate where a problem may occur and already delivers information to him.

3.3.4 Future work

We have suggested a proactive, user-oriented QoS management approach. While the proactive part consists of methods of observations and predictions of the users behaviour and expectations, the user oriented approach provides us with metrics and the monitoring resource that form the basis for a provision of QoS.

Identification of user-oriented QoS parameters, the influence of context information and usability issues as well as forecasting and proactive control mechanisms are the key aspects to be considered and need to be further elaborated. We tried to sketch in this contribution seminal work that this research work is based on. Future work will focus on refining the general architecture and specifying in detail its components and interfaces. The design and specification will be accompanied by prototypical implementation to demonstrate the feasibility of the suggested approach.

3.4 Measuring the QoS of a Satellite Based Content Delivery Network

Helmut Hlavacs
University of Vienna, Austria

3.4.1 Introduction

Today the most cost-effective way for transporting data between professionals being apart thousands of kilometers is to use the Internet and some kind of encryption, for instance by using a virtual private network (VPN) or an extension of established standards like SSL or HTTPS. This approach however has proven to comprise uncalculated risks, as the Internet itself interconnects everybody to everybody without distinction, and, simply by being connected to the Internet, malevolent people or programs may easily reach any target computer attached to the Internet.

The IST project CODIS (COntent Delivery Improvement by Satellite), which forms up a closed content delivery network (CDN) interconnecting four European sites. As a backbone, CODIS uses a high-speed satellite link. The purpose of CODIS is to demonstrate the usefulness of such a CDN for various industries including content providers, broadcasters, internet service providers (ISPs), and multimedia application end users. The CODIS consortium, consisting of Alcatel Space, the French space agency CNES, the broadcasting research institutions Télédiffusion de France (TDF) and Institut für Rundfunktechnik (IRT), the measurement equipment manufacturer Rohde & Schwarz, the content management system provider Activia, and the Institute for Computer Science and Business Informatics of the University of Vienna, has setup, run and evaluated a satellite based CDN using the satellites Telecom 2D and Atlantic Bird 2, both situated at 11 degree West.

Figure 3.4 shows the CODIS fully meshed network. The sites in Toulouse, Metz and Munich are able to send and receive, while the Vienna side is in receive mode only, the main purpose of the Vienna side is to observe, measure and interpret the quality of service from the end user point of view. The architectures of Toulouse and Vienna resemble an ISP like setup, while due to their DVB-T stations, Metz and Munich additionally cover the networks of broadcasters.

3.4.2 The QoS measurement framework

For measuring the QoS of the CODIS network we first analysed the CODIS protocol stack that is used in the satellite network, but also in the wireline networks being attached to CODIS [36]. The protocol layers under consideration are shown in Figure 3.5.

For each of these layers we then defined

- QoS metrics to be measured,

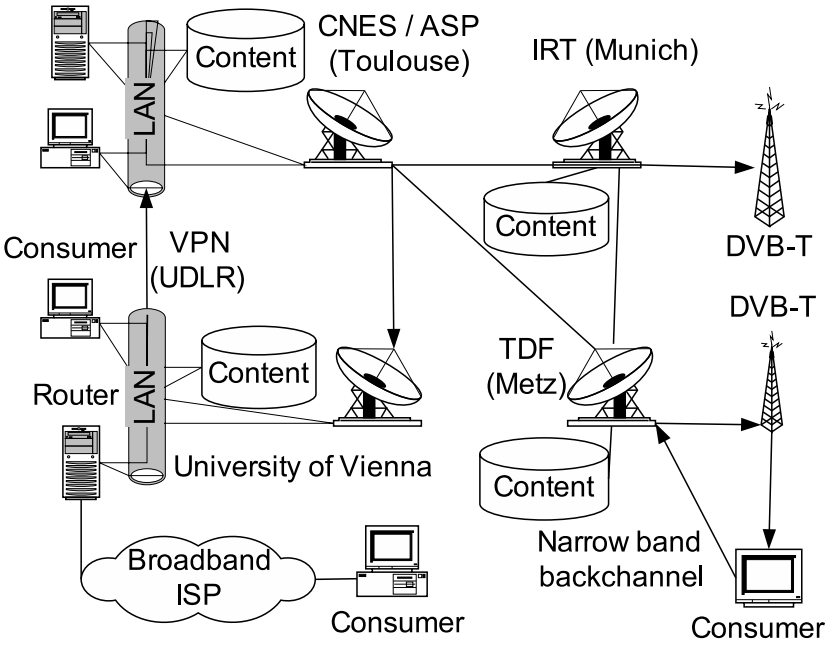


Figure 3.4: The CODIS fully meshed network.

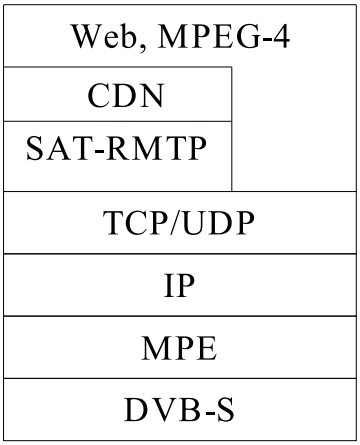


Figure 3.5: The CODIS protocol stack.

Table 3.1: The CODIS QoS metrics.

Protocol Layer	Metric
Application	Response time, video picture quality, frame rate,
	bitrate, loss rate, prob. for frame loss/stalling
CDN	Publication time, cache hit rate, bandwidth out of cache, CPU/disk/network usage
IP	Hop count, network latency, round trip time, bottleneck bandwidth, loss probability, bulk transfer capacity, link latencies and bandwidth, jitter, conditional loss probability, loss gap
DVB-S	Follows [37]

- the applications that generate these metrics,
- Measurement equipment and software, and
- if possible, a subjective QoS interpretation of the measurement results.

The chosen QoS metrics can be seen in Table 3.1.

As target applications we focussed on multimedia traffic created by Web pages and MPEG-4 video streaming.

3.4.3 Measurement results

The measurement results are presented in [9]. The conclusion of the QoS measurements for CODIS are:

- High bit rate (2 Mbit/s +)
- High round trip time / latency
- Good for data casting / CDN / multicasting
- Not good for video telephony / conferencing
- Satellite yields reliable, stable backbone, though packet losses occur
- Many interactive applications (e.g. Web) yield sufficiently low response time

3.5 Pseudo-subjective video and audio quality

Samir Mohamed, Gerardo Rubino, and Martín Varela
IRISA, Rennes, France

This subsection describes an approach developed at INRIA, Unit of Rennes (IRISA), whose aim was to provide a tool of quality evaluation for multimedia flows with the following characteristics:

- it evaluates the quality of a video or audio flow *numerically* (for instance, within a MOS (Mean Opinion Score) scale);
- it works automatically and, if necessary, in real time; this means two things:
  - the method is not computationally intensive;
  - and there is no need to access the original multimedia signals (before encoding and transmission); our method, when in operation, only works with the received signal;
- it evaluates the flow quality *as perceived* by the human observer at the terminal side, which means that it gives to the flow a numerical value close to the value that a set of humans performing a well controlled subjective evaluation would give to the flow (the reason for this is that our procedure is partially built using subjective data);
- it allows to analyze the quality as a function of many factors, either related to the source or to the network.

In the sequel we describe in more detail our approach and its first applications.

3.5.1 Our approach: Pseudo-subjective Quality Assessment

The method used here [38, 39] is a hybrid between subjective and objective evaluation, which can be applied to speech, high-quality audio and even video. The idea is to have several distorted samples evaluated subjectively, and then use the results of this evaluation to teach a Random Neural Network (RNN) the relation between the parameters that cause the distortion and the perceived quality. In order for it to work, we need to consider a set of P parameters (selected *a priori*) which may have an effect on the perceived quality. For example, we can select the codec used, the packet loss rate of the network, the end–to–end delay and/or jitter, etc. Let this set be $\mathcal{P} = \{\pi_1, \ldots, \pi_P\}$. Once these *quality–affecting* parameters are defined, it is necessary to choose a set of representative values for each π_i, with minimal value $\pi_{\min}$ and maximal value $\pi_{\max}$, according to the conditions under which we expect the system to work. Let $\{p_{i1}, \cdots, p_{iH_i}\}$ be this set of values, with $\pi_{\min} = p_{i1}$ and $\pi_{\max} = p_{iH_i}$. The number of values to choose for each parameter depends on the size of the chosen interval, and on the desired precision. For example, if we consider the packet loss rate as one of the parameters, and if we expect its values to range mainly from 0 %

to 5 %, we could use 0, 1, 2 and 5 % as the selected values, or in a more conservative way, the set $\{0\%, 1\ \%, 2\ \%, 3\ \%, 5\ \%, 10\ \%\}$. In this context, we call *configuration* a set with the form $\gamma = \{v_1, \ldots, v_P\}$, where v_i is one of the chosen values for p_i.

The total number of possible configurations is usually very large. For this reason, the next step is to select a subset of the possible configurations to be subjectively evaluated. This selection may be done randomly, but it is important to cover the points near the boundaries of the configuration space. It is also advisable not to use a uniform distribution, but to sample more points in the regions near the configurations which are most likely to happen during normal use. Once the configurations have been chosen, we need to generate a set of "distorted samples", that is, samples resulting from the transmission of the original media over the network under the different configurations. For this, we use a testbed, or a network simulator.

Formally, we must select a set of M media samples (σ_m), $m = 1, \cdots, M$, for instance, M short pieces of audio (subjective testing standards advise to use sequences having an average 10 sec length). We also need a set of S configurations denoted by $\{\gamma_1, \cdots, \gamma_S\}$ where $\gamma_s = (v_{s1}, \cdots, v_{sP})$, v_{sp} being the value of parameter π_p in configuration γ_s. From each sample σ_i, we build a set $\{\sigma_{i1}, \cdots, \sigma_{iS}\}$ of samples that have encountered varied conditions when transmitted over the network. That is, sequence σ_{is} is the sequence that arrived at the receiver when the sender sent σ_i through the source-network system where the P chosen parameters had the values of configuration γ_s.

Once the distorted samples are generated, a subjective test (e.g. as in ITU P.800 recommendation for speech) is carried out on each received piece σ_{is}. After statistical processing of the answers, the sequence σ_{is} receives the value μ_{is} (often, this is a *Mean Opinion Score*, or MOS). The idea is then to associate each configuration γ_s with the value

$$\mu_s = \frac{1}{M} \sum_{m=1}^{M} \mu_{ms}.$$

At this step we have a set of S configurations $\gamma_1, \ldots, \gamma_S$. Configuration s has value μ_s associated with it. We randomly choose S_1 configurations among the S available. These, together with their values, constitute the "Training Database". The remaining $S_2 = S - S_1$ configurations and their associated values constitute the "Validation Database", reserved for further (and critical) use in the last step of the process.

The next step is to train a statistical learning tool (in our case, a RNN) to learn the mapping between configurations and values as defined by the Training Database. Assume that the selected parameters have values scaled into $[0, 1]$ and the same with quality. Once the tool has "captured" the mapping, that is, once the RNN is trained, we have a function $f()$ from $[0, 1]^P$ into $[0, 1]$ mapping now any possible value of the (scaled) parameters into the (also scaled) quality metric. The last step is the validation phase: we compare the value given by $f()$ at the point corresponding to each configuration γ_s in the Validation Database to μ_s; if they are close enough for all of them, the RNN is validated (in Neural Network Theory, we say that the tool *generalizes well*). In fact, the results produced by the RNN are generally closer to the MOS than that of the human subjects (that is, the error is less than the average deviation between human evaluations). As the RNN generalizes

well, it suffices to train it with a small (but well chosen) part of the configuration space, and it will be able to produce good assessments for any configuration in that space. The choice of the RNN as an approximator is not arbitrary. We have experimented with other tools, namely Artificial Neural Networks, and Bayesian classifiers, and found that RNN perform better in the context considered. ANN exhibited some problems due to over-training, which we did not find when using RNN. As for the Bayesian classifier, we found that while it worked, it did so quite roughly, with much less precision than RNN. Besides, it is only able to provide discrete quality scores, while the NN approach allows for a finer view of the quality function.

The neural network model used has some interesting mathematical properties, which allow, for example, to obtain the derivatives of the output with respect to any of the inputs, which is useful for evaluating the performance of the network under changing conditions (see next section).

The method proposed produces good evaluations for a wide range variation of all the quality affecting parameters, at the cost of one subjective test.

3.5.2 Performance of our approach on the case of speech

In this section we present the results we obtained with our approach for two different VoIP test campaigns we have performed.

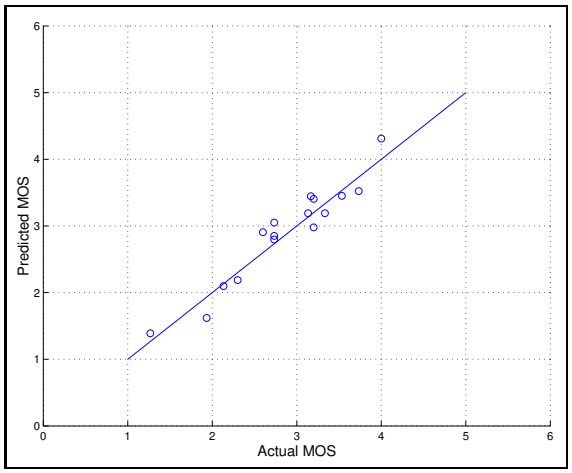
For the first battery of tests we considered the parameters listed in Table 3.2.

Table 3.2: Network and encoding parameters and values used for the first test set.

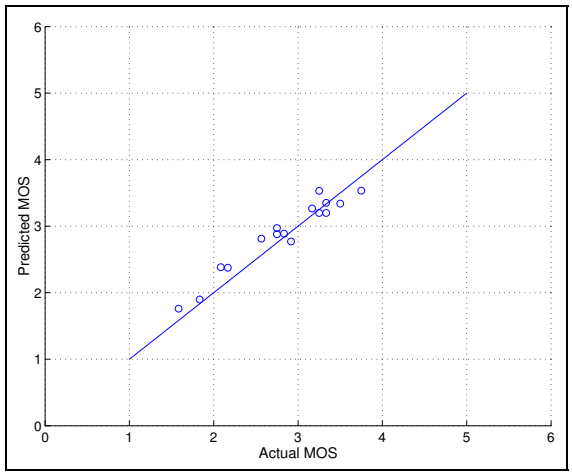
Parameter	**Values**
Loss rate	0 %...40 %
Loss burst size	1...5
Codec	PCM Linear-8, G.726 and GSM-FR
Packetization interval	20, 40, 60 and 80 ms

With these parameters, we simulated the network effects on encoded files, and used these files to conduct MOS tests (as specified in ITU P.800 Rec. in three languages: French, Spanish and Arabic). Once the MOS results were screened, we proceeded as described in Section 3.5.1, and trained three RNN, one for each language considered. The results obtained were very good, with correlation coefficients of 0.99 for Spanish and Arabic, and 0.98 for French (using only validation data). Figure 3.6 shows scatter plots for these tests.

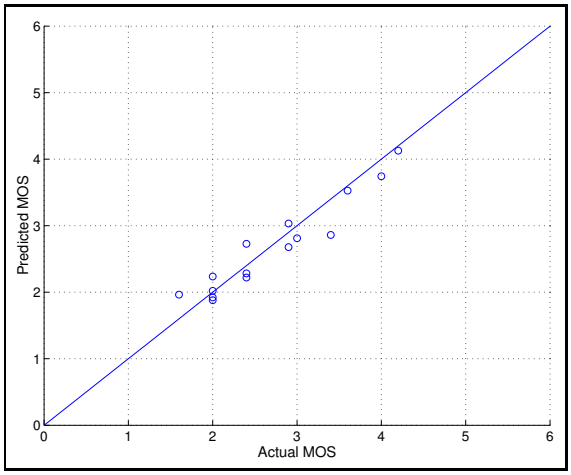
For the second set of tests, we refined our network model using a simplified Gilbert model, see [39] for more details. The distorted speech samples were generated on a live network using the Robust Audio Tool (RAT) and a proxy that generated the losses as specified on a live network. A MOS test was performed and the results screened. We tested several RNN architectures and various combinations of training/validation database sizes, and found good results using about 100 samples for training, and 10 for validation. We also considered Forward Error Correction (FEC) parameters in these tests. The parameters



(a) Spanish samples – Correlation Coefficient = 0.99



(b) Arabic samples – Correlation Coefficient = 0.99



(c) French samples – Correlation Coefficient = 0.98

Figure 3.6: Scatter plots for the first series of tests. Estimations are for validation data (never seen before by the RNN).

considered for our experiment are listed on table 3.3. The results obtained varied with the different sizes of training/validation databases, and yielded correlation coefficients between 0.73 and 0.93 with actual MOS values. It is interesting to see that even when using relatively small sets of training samples, very good results can be obtained, and this allows for a trade-off between cost and performance for our method (since its main cost is that of performing the subjective tests to train the RNN). Figure 3.7 shows a scatter plot for the validation data of the second set of tests.

To end this section, let us just comment that we recently proposed an application of our technology for performance evaluation purposes. The idea is to couple our evaluation tool with a classical model (in our first example, a classical queueing model) and then to relate

Table 3.3: Network and encoding parameters and values used for the second test set.

Parameter	Values
Loss rate	0 %...15 %
Mean loss burst size	1...2.5
Codec	PCM Linear 16 bits, GSM
FEC	ON(GSM)/OFF
FEC offset	1...3
Packetization interval	20, 40, and 80 ms

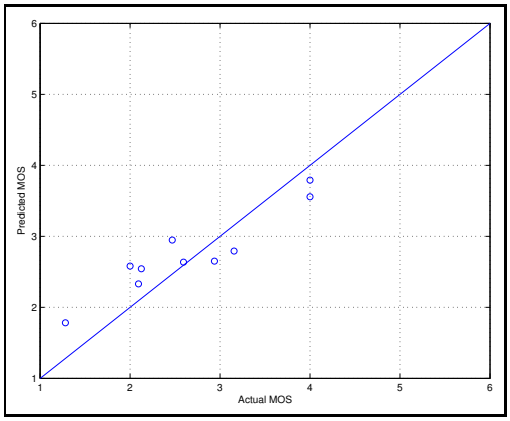


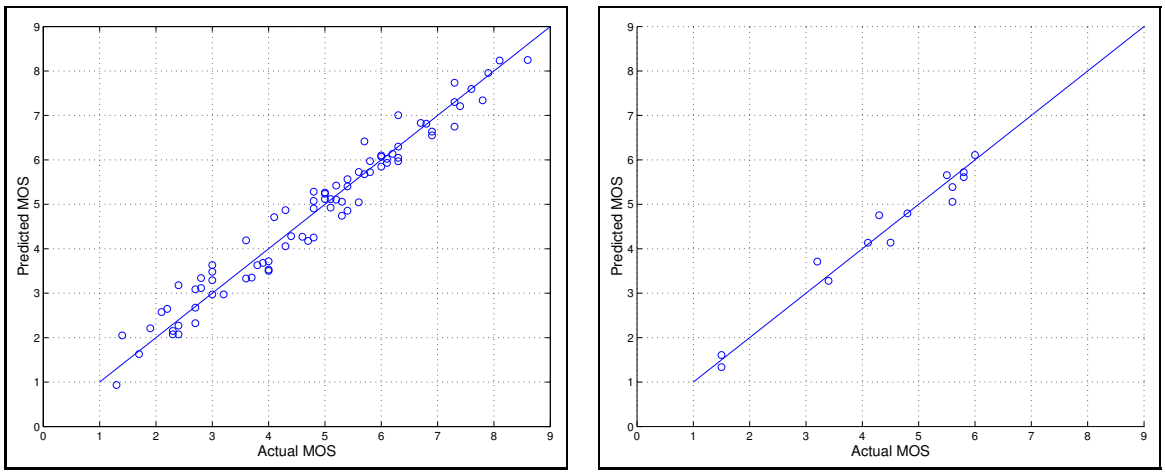
Figure 3.7: Scatter plot for the second series of tests – Correlation Coefficient = 0.93. Estimations are for validation data (never seen before by the RNN).

perceived quality to load parameters (such as offered traffic, link speeds, etc.). See [40] for more details on this.

3.5.3 Performance of our approach on the case of video

To generate the distorted video sequences, we used a tool that encodes a real-time video stream over IP networks into the H.263 format, simulates the packetization of the video stream, decodes the received stream, and allows us to simulate the network transmission conditions (packet loss process, etc.). The encoder can also be parameterized, in order to control the bit rate, the frame rate, the intra macro blocs refresh rate (i.e. it encodes the given macro bloc into intra mode rather than inter mode – this is done to make the stream more resistant to losses). Thus, the considered parameters are the bit rate, the frame rate, the error resilient factor, the loss rate and the loss burst size. We generated a total of 94 distorted video sequences in CIF format. Subjective test is carried out based on the ITU-R BT.500 Rec [13]. After carrying out the MOS experiment for the generated 94 samples, we divided our database into two parts: one to train the RNN containing 80 samples, and the other to test the RNN's accuracy to work in a dynamic environment, containing 14 samples. After training the RNN and comparing the training data against the values predicted by the RNN, we got a correlation coefficient = 0.99. The results are shown in Fig. 3.8(a). When the tesing database was applied to RNN, we obtained a

correlation coefficient of 0.98, see Fig. 3.8(b).



(a) Training DB

(b) Testing DB

Figure 3.8: Scatter plots showing the correlation between Actual and Predicted MOS scores in the case of video.

3.6 Using Throughput Statistics for End-to-End Identification of Application-Perceived QoS Degradation

Markus Fiedler, Patrik Carlsson
Blekinge Institute of Technology, Karlskrona, Sweden

Kurt Tutschku
University of Würzburg, Germany

3.6.1 Motivation

Users of advanced, distributed applications have a somewhat ambiguous relationship to the networks: they need them, but they should not feel their presence at all. In particular in packet switched networks, all the packets should appear at the other side with more or less the same timing relationships as they were sent into the network. In such a case, the application-perceived QoS would be perfect.

Fixed packet delay, of course, is unavoidable due to physical constraints. Stochastic variation of the packet delay in the network or packet loss, however, are the typically results of data streams contenting for common resources in a best-effort manner. Streams of packets which traverse *bottlenecks*, i.e. locations in the network of temporary or permanent shortage in capacity [41], experience a significant change of their statistical characteristics. The *type of change* depends on the *nature of bottlenecks* which are passed. Packet streams which content in bottleneck experience some kind of *sharing* behavior due to multiplexing, whereas streams which pass limiting bottlenecks suffer a *shaping* behavior. In addition, the strength of the change characteristic reflects the *severity* of QoS degradation due to bottlenecks.

From the viewpoint of the application, both type and severity of the change are of high importance. A first step is to identify the changing behavior and to visualize the behavior in order to relate it to the perceived performance of the application.

In the sequel, we present a way of identifying the change behavior of bottlenecks based on measurements of throughput statistics at both the sending and the receiving side of the network [42].

3.6.2 Throughput histogram difference plots

The performance parameter *throughput* has been chosen as the basis for identifying the change behavior since it combines performance problems on packet level with application and user perception. Delays and losses lead to a reduction of throughput, and the application perceives the network being "slow" and "lossy". The user in turn has to wait unnecessarily long for a transaction to be finished or a file to be downloaded – or even face the fact that a service breaks down.

The traditional notion of throughput, as the average of speed received for a complete flow of packets, is extended to a short-term variant measured in comparably small *averaging interval* of duration ΔT (typically between 100 ms and 1 s) during a time window or *observation interval* of duration ΔW (typically in the order of minutes). Thus, we obtain $n = \Delta W/\Delta T$ values of a throughput time series $\{R_s\}_{s=1}^{n}$. For a packet stream of interest, throughput measurements are carried out at the ingress and egress of the network, starting with the first packet of that particular stream. It is important to note that no advanced clock synchronization is required. [42] details how to derive the throughput time series $\left\{R_s^{\text{in}}\right\}_{s=1}^{n}$ and $\{R_s^{\text{out}}\}_{s=1}^{n}$ from packet traces; the same principle is applicable to on-line calculations. As we are interested in the change characteristic of the whole network path as perceived by the application, the network as such is treated as some kind of "black box". However, the method can be applied to whatever potential bottleneck in the network, given that packet streams can be observed at both entrance and exit of that particular bottleneck. In general, the method reveals the experience of a packet stream, which is the viewpoint of an application and the user, towards the network behavior.

From the comparison of the time series $\left\{R_s^{\text{in}}\right\}_{s=1}^{n}$ and $\{R_s^{\text{out}}\}_{s=1}^{n}$, we can deduct the impact of the bottleneck on the perceived throughput. Such a direct comparison is generally possible, but practically unfeasible in particular when the length of the time series n gets large. Thus, we focus on a condensed representation in form of throughput histograms $\mathcal{H}\left(\left\{R_s^{\text{in}}\right\}_{s=1}^{n}, \Delta R, \Delta T, \Delta W\right)$ and $\mathcal{H}\left(\{R_s^{\text{out}}\}_{s=1}^{n}, \Delta R, \Delta T, \Delta W\right)$ with a throughput resolution of ΔR (cf. [42] for the corresponding formulas). A predecessor work [41] has shown that from comparing such histograms, information on the existence and type of a bottleneck can be derived. Empirical studies have shown that the comparison works well for about $\lceil R_{\max}/\Delta R + 1 \rceil \simeq 20$ intervals for $n \geq 600$. Such compact histograms can much easier be transferred between receiver and sender than the original time series.

The comparison of throughput histograms as such happens through calculating *throughput histogram difference plots* $\Delta\mathcal{H}\left(\{R_s^{\text{out}}\}_{s=1}^{n}, \left\{R_s^{\text{in}}\right\}_{s=1}^{n}, \Delta R\right)$. Negative (positive) values in these plots reveal that a certain speed is less (more) frequent at the outlet as compared to the inlet. From such speed changes, we can deduct what happens with the packet stream on its way through the network.

3.6.3 Types of bottlenecks

Figure 3.9 depicts a typical bottleneck scenario. The packet streams of a video application passes through a single link, which has maximum capacity of 10 Mbps. The video data stream contents on this resource with a constant bitrate cross traffic, i.e. a disturbing packet stream.

Figure 3.10 shows throughput histogram plots from the measurement of the video conference application for various speeds of the disturbing cross traffic.

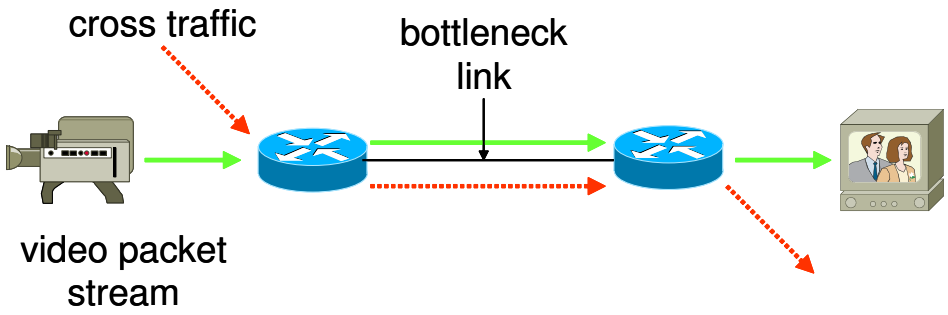


Figure 3.9: Bottleneck scenario.

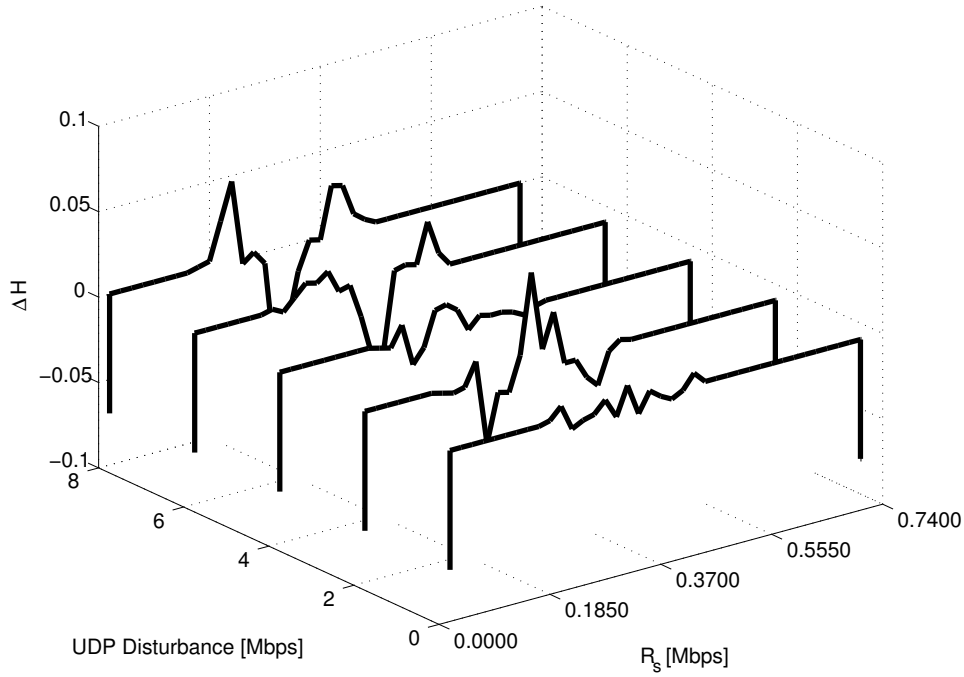


Figure 3.10: Throughput histogram difference plots for video from Karlskrona to Würzburg for different levels of disturbance in a local bottleneck.

Shaping bottleneck

At a low level of disturbance, i.e. 2 Mbps of cross traffic, we observe negative $\Delta\mathcal{H}$ values for both low and high speeds, but positive $\Delta\mathcal{H}$ values for typical speeds. Thus, the packet traffic at the output flows more regularly than at the input: The bottleneck behaves as shaper towards a typical speed. The throughput histogram difference plot looks roughly like a "W". In this specific case, no impact on the video quality was perceived by the users.

Shared bottleneck

At a comparably high level of disturbance (6 to 8 Mbps), we now observe some kind of an inverted shape of the throughput histogram difference plots as compared to before. They roughly look like an "M" with positive $\Delta\mathcal{H}$ values for both low and high speeds and negative $\Delta\mathcal{H}$ values for typical speeds. The increase in low speeds stems from building a queue, while the increase in high speeds reflects dequeuing at the outlet of the bottleneck [41].

While the user does not feel any quality degradation at 6 Mbps disturbance, the user does at 8 Mbps. The severity of the change is reflected in the minimum and maximum values of the differences in the plot. It seems that the severity has surpassed a critical threshold from which on the application itself cannot cope with the network QoS problems any more.

Overloaded bottleneck

Not visible in Figure 3.10 is the case when the bottleneck is overloaded, i.e. the average input exceeds the average capacity. In that case, traffic can hardly be buffered any more, but gets lost. The shape of the corresponding throughput histogram difference plots becomes an "N". The negative $\Delta\mathcal{H}$ values indicates speeds the bottleneck cannot cope with, while the positive $\Delta\mathcal{H}$ values reveal the typical speed supported by the bottleneck. Due to persisting overload and a limited buffer, the data stream experiences considerable loss. Probably due to this fact, the video conference "died" at 10 Mbps disturbance.

Undetermined bottleneck

In contrary to the already presented cases, the plots of the remaining cases (disturbance of 0 and 4 Mbps, respectively) do not reveal the type of bottleneck in a clear way. It is important to note that even the 0 Mbps case displays the existence of changes; however, these are quite small as compared to the situations described above. The 4 Mbps case introduces larger changes; the shape of the throughput histogram difference plot seems to be a mixture of "W" and "M", which is no surprise as the type of the bottleneck changes when increasing the disturbance from 2 Mbps to 6 Mbps.

3.6.4 Ongoing and future work

In a recent Master's thesis, jointly advised by the Blekinge Institute of Technology and the University of Würzburg, a "real bottleneck" (a serial link between two routers with tuneable bit speed) was investigated, including parameter studies of time and throughput resolution ΔT and ΔR, respectively.

Interesting topics to study in the future (e.g. within JRA.6.1) include:

- How to define thresholds for different applications?
- How general is the indicator in terms of applications?
- To what extent is automatic recognition of change patterns in data streams possible?
- In which way can such a performance indicator provide feed-back to customers and/or control algorithms about the network QoS?

3.7 User Utility Functions for Auction-based Resource Reservation in 2.5/3G Networks

Manos Dramitinos, George D. Stamoulis, and Costas Courcoubetis
Research Center – Athens University of Economics and Business, Greece

3.7.1 Motivation – the problem

Multi-unit auctions have recently received considerable attention as an economic mechanism for resource reservation in networks. The case where users compete for reserving *consistently* resources for large time scales remains an open research topic. This is of particular interest for many practical cases involving the provision of network services with relatively high duration. A prominent case is that of UMTS [43]: users request services for large time scales, e.g. several minutes in order to watch video clips at their terminal; on the other hand, the duration of network slots t_a, over which resource units can be allocated[2], is much shorter. The fact that the population of users generally varies over time further complicates the problem. Apart from UMTS, the open problem of consistent resource reservation also applies to GPRS technology (including its enhanced version EDGE) [44]. We propose a series of consecutive auctions (of a certain type) as a means for attaining efficiently consistent reservation of resources. Since constant resource allocation may not be feasible for all users and no strict Quality of Service (QoS) guarantees are provided, it is of great importance to construct *meaningful user utility functions* that actually express users' preferences. These utility functions are additive so that they can be used as bidding functions in our mechanism, thus providing a quantification of users' preferences in cases of inconsistent resource allocation patterns.

3.7.2 ATHENA: A new resource reservation mechanism

Our approach for UMTS resource reservation is called ATHENA (Auction-based THird gEneration Networks resource reservAtion) and consists in conducting a sequence of "mini-auctions" of the short time scale t_a of slots. Each mini- auction is a sealed-bid auction with *atomic* bids (i.e. bids that are either fully satisfied or rejected) of the type (p, q), where p is the expressed willingness to pay for a quantity q of resource units in the present slot. (We comment on the number of such bids permissible per user in the next subsection.) For UMTS, if the service is of a specific rate m, then we have $q = m \cdot t_a$. Each user is charged with the social opportunity cost that his presence entails; that is, each mini-auction is a Generalized Vickrey Auction [45].

However, in a realistic case of a UMTS network, it is not feasible for users to participate in all these mini-auctions, either manually or automatically by means of an agent running

[2]The unit of resource allocation and the definition of a slot depends on the network technology. In UMTS, which supports provision of bandwidth on demand (BoD), resources are allocated in quantities of bits to be transported within a 10msec UTRAN frame; hence, for UMTS, the unit coincides with *one bit*. In GPRS, for which the work to be presented is also applicable, the unit of resource allocation is the *radio block*.

in their respective terminal. Thus, since the user cannot give his utility on a per mini-auction basis, we define meaningful utility functions, pertaining to the various services. These functions are provided by the network operator for the user to choose from and scaled by the user's total willingness to pay, which is to be given by the user himself. Then, the network runs all mini-auctions by *bidding optimally* (i.e., truthfully) *on behalf of each of the users*, according to his respective selection of utility function. Thus, *all computation is performed on the network-base station* rather than on the user terminals. The network and auction complexity are hidden from the users: A user demanding a service selects among the predefined utility functions the one that better expresses his preferences and declares a willingness to pay U; a session that lasts for time t_s is then created. Each user aims in achieving constantly the desired rate m by bidding in a large number K_s of mini-auctions, where $K_s = \frac{t_s}{t_a}$. (Recall, however, that the network is bidding on each users behalf.) If the user wishes to watch his favorite music video clip lasting for 4 minutes, all that he declares is the video name, the desired quality level, the total willingness to pay U_s, and the utility function type. The parameters t_s, K_s and $m = 2Mbps$ are computed automatically by the network and are transparent to the user.

3.7.3 User utility functions

We assume that the user's value for obtaining the service u_s is the sum of the marginal "sub-utilities" attained due to each successful allocation; thus, $u_s = u(x_1, ..., x_{K_s}) = \sum_{i=1}^{K_s} u_i$. Next, we define *meaningful utility functions, pertaining to the various services.* These functions reflect the fact that, when there are gaps in the resource allocation pattern, not only the amount of slots but also the *way* these are allocated makes considerable difference to the degree of user satisfaction. Thus, by selecting one of the predefined user utility functions, each user declares his preferred form of allocation pattern for the cases where perfectly consistent resource reservation is not possible. Hence, these functions accurately express the value attained from the service, from a user perspective. In particular, we have defined the following three user types and the corresponding utility functions:

- Type 1: *Indifferent to the allocation pattern.* This applies to volume-oriented users, such as those downloading news articles. The utility attained depends only on the quantity allocated, as opposed to the allocation pattern; hence, $u_i = \mathbf{1}(x_i = m) \cdot \frac{U_s}{K_s}$.
- Type 2: *Sensitive to the service continuity.* This type pertains (among other cases) to users that prefer watching consistently half of a football match rather than watching multiple shorter periods. Thus, they prefer the allocation pattern of Figure 3.11(a) to that of Figure 3.11(b). In order to express this preference, we define the sub-utility function to be $u(x_i; h_{i-1}) = \mathbf{1}(x_i = m)\frac{U_s}{K_s} \cdot \alpha^d$ where d is the distance between the current and the previous slots during which this user achieved reservations; h_{i-1} is the history of resource allocation for this user up to the present slot, and influences $u(x_i; h_{i-1})$ through the value of d, which is kept track of by the ATHENA module.
- Type 3: *Sensitive to the smoothness of the allocation pattern.* This is the case for stock-market information. Such customers prefer allocation pattern Figure 3.11(b) to that of Figure 3.11(a). The corresponding sub-utility is $u(x_i; h_{i-1}) = \mathbf{1}(x_i =$

$m)\frac{U_s}{K_s} \cdot \alpha^{\max\{0,\Delta d\}}$ where Δd is the difference of the present and the previous values of the distance defined above. Note that the $\alpha^{\max\{0,\Delta d\}}$ equals 1 if Δd is negative, and thus the received quality of service improves or remains constant, and is less than 1 (since $0 < \alpha < 1$) if the distance increases and hence the quality deteriorates.

Both Type 2 and 3 users have certain features in common: coefficient $\frac{U_s}{K_s}$ expresses the user satisfaction according to the number of units allocated, while α and its powers declare the (dis-)satisfaction resulting from the gaps in the allocation pattern. In both cases, if the user is constantly allocated resources (and thus the best possible quality is achieved), then the utility obtained is U_s. Note that, due to the fact that the network is bidding on behalf of the users, the *incentives* for each user only concern his selection of one of the predefined utility functions and his declaration of the total willingness to pay U_s for this service. The incentive compatibility property shows that a user whose preferences are accurately expressed by one of the predefined functions, has the incentive to *truthfully* declare this function as well as U_s.

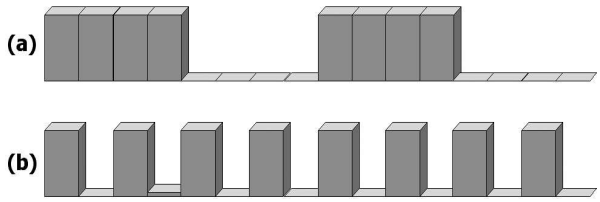


Figure 3.11: Patterns with inconsistent resource allocation: although the two patterns lead to the same mean rate, different users may prefer one of them to the other.

The aforementioned utility functions are not the only ones reflecting the user satisfaction w.r.t. the allocation pattern attained. What is important, is the fact that the values of these utilities reflect correctly the preferences of each type of user. We have extended the definitions of these utility functions so as to cover more interesting cases. For instance, a user may be willing to watch just "good quality" video - of a rate r_{good} - whenever watching the video with the preferred "high quality" is not feasible. Watching the video with consistently either "good quality" or "high quality" results in different degrees of user satisfaction; hence, the total willingness to pay respectively equals V_{good} and $V_{\text{high}} = V_{\text{good}} + \Delta V$ where ΔV expresses the user's *extra* satisfaction for video of "high quality". Due to the possible fluctuations of the rate attained, a proper utility function for this type of users is $u(x_i; h_{i-1}) = \mathbf{1}(x_i \geq r_{\text{good}})\frac{V_{\text{good}}}{K_s} \cdot \alpha^{d_1} + \mathbf{1}(x_i = r_{\text{high}})\frac{\Delta V}{K_s} \cdot \alpha^{d_2}$, where d_1 and d_2 are defined w.r.t. the length of the gaps incurred in the r_{good} and $r_{\text{high}} - r_{\text{good}}$ allocation sub-patterns respectively. Finally, note that the number of atomic bids to be given on behalf of each user at each mini-auction equals the number of alternative quality levels. Thus, for the aforementioned case of two such levels, two *summable* bids should be given per user: one expressing his willingness to pay for the basic rate r_{good} and the other expressing his extra willingness to pay for the extra rate $r_{\text{high}} - r_{\text{good}}$. Of course, in the simple case of a single quality level that was discussed in the previous paragraphs, only one bid is to be submitted per user in each mini-auction.

3.7.4 Conclusions and further work

We conclude that our approach implements dynamic pricing and optimal resource allocation in a UMTS network, ensuring that users both receive meaningful service and are charged in a fair manner, according to the actual demand for network resources as expressed by the users themselves. A mapping of the aforementioned utility functions to the UMTS service classes is to be provided as well. It is worth noting that the application perceived QoS and the user perceived QoS are coupled and "re-engineered" in our approach. This stems from the way the user utility functions are constructed and their usage as bidding functions by taking into account the resource allocation patterns. Since the auction bids are computed via these functions, it is clear that they also affect future resource allocations, hence the network resource reservations.

3.8 A Moving Average Predictor for Playout Delay Control in VoIP

Víctor M. Ramos R., Chadi Barakat, and Eitan Altman
INRIA, France

3.8.1 Introduction

In this work we propose an algorithm for playout delay adaptation with tunable loss rate. We focus on the tradeoff between loss and delay for playout delay control algorithms in VoIP. Using measurements of packet end-to-end delay of audio sessions done with NeVoT, we present and validate a Moving Average (MA) algorithm that adjusts the playout delay at the beginning of each talkspurt. To prove the efficiency of our algorithm, we compare it with earlier work done by Ramjee et al. [46]. We present two versions of our algorithm: an offline algorithm and an online one. The offline MA algorithm serves as a reference for our work. Then, we show how an online hybrid algorithm can be implemented by combining the ideas proposed by Ramjee et al. with the moving average algorithm we propose.

One characteristic that most of the playout delay adaptation algorithms lack is the ability to fix the loss percentage to some a priori value. This characteristic would allow to control an important QoS parameter, the late loss rate, as perceived by a user. By changing a measure of variability, the algorithms proposed by Ramjee et al. can achieve different loss percentages. However, there is no explicit relationship between the measure of variability that we can adapt in these algorithms and the average loss percentage. The average loss percentage can change from one audio session to another, even if this parameter is kept unchanged. Here lies the main contribution of our work. The moving average algorithm we propose adjusts the playout delay from talkspurt to talkspurt, given a desired target of average loss percentage p. Our algorithm ensures that the average loss percentage we obtain during the session is close, if not equal, to the target value. At the same time, and in most of the cases, our algorithm realizes this target with a smaller average playout delay than the one we need to obtain the same average loss percentage with the algorithms proposed by Ramjee et al. For practical loss percentages, we validate our algorithm and those of Ramjee et al. using real packet audio traces. By using collected audio traces we can compare the algorithms under the same network conditions. This work has been presented and published in [47].

Table 3.4 shows the notation we use in this section.

3.8.2 Performance measures

To assess the performance of a playout adaptation algorithm, we focus on the total number of packets that are played out during an audio session, as well as on the experienced average end-to-end delay. Suppose we are given a packet audio trace with the sender and receiver timestamps of audio packets. Let p_k^i, N, L, N_k, t_k^i, and a_k^i be defined as in Table 3.4. r_k^i indicates if packet i of talkspurt k is played out or not. So, r_k^i is defined as:

Table 3.4: Definition of variables.

Param.	Meaning
L	The total number of packets arriving at the receiver during a session.
N	The total number of talkspurts in a session.
N_k	The number of packets in talkspurt k.
t_k^i	The time at which the i-th packet of talkspurt k is generated at the sender.
a_k^i	The time at which the i-th packet of talkspurt k is received.
d_k^i	The variable portion of the end-to-end delay of the i-th packet in talkspurt k. $d_k^i = a_k^i - t_k^i - \min_{\substack{1\leq k\leq N \\ 1\leq i\leq N_k}} (a_k^i - t_k^i)$.
p_k^i	The time at which packet i of talkspurt k is played out.

$$r_k^i = \begin{cases} 0, & \text{if } p_k^i < a_k^i \\ 1, & \text{otherwise.} \end{cases}.$$

The total number of packets, T, played out in an audio session is thus given by:

$$T = \sum_{k=1}^{N} \sum_{i=1}^{N_k} r_k^i. \tag{3.1}$$

The average playout delay, D_{avg}, is equal to :

$$D_{avg} = \frac{1}{T} \sum_{k=1}^{N} \sum_{i=1}^{N_k} r_k^i [p_k^i - t_k^i]. \tag{3.2}$$

Finally, the loss percentage, l, is equal to :

$$l = \frac{L - T}{L} \times 100. \tag{3.3}$$

3.8.3 Moving Average prediction

The model

Let D_k be the optimal playout delay at the beginning of talkspurt k, and let p be the desired average loss percentage per-session. We mean by *optimal playout delay* the playout delay that makes the number of losses per talkspurt the closest to $p \times N_k$, N_k being the number of audio packets received during the k-th talkspurt. By controlling the loss percentage per-talkspurt to p, we are sure that the overall loss percentage during the whole

audio session is also close to p. We compute D_k as follows, let d_k^j be the variable portion of the end-to-end delay of the j-th packet in talkspurt k. For each talkspurt, $1 \leq k \leq N$, we sort in ascending order the packet end-to-end delay values to obtain N new ordered sets $\{d_{k_{sort}}^j\}$, with $1 \leq j \leq N_k$. We set the optimal playout delay of the k-th talkspurt to the following value:

$$D_k = d_{k_{sort}}^i, \quad i \leq N_k, \tag{3.4}$$

with $i = \texttt{round}((1-p)N_k)$. Thus, if $d_k^i \leq D_k$, the i-th packet of talkspurt k is played out, otherwise the packet is dropped due to a late arrival.

Consider that we have a set of optimal delay values in the past $\{D_k, D_{k-1}, D_{k-2}, \ldots\}$, and that we want to predict the value of D_{k+1}. The predicted value of D_{k+1} is denoted by $\hat{D}_{k+1}$, and is taken as a weighted average of the last M values of the process $\{D_k\}$. Thus,

$$\hat{D}_{k+1} = \sum_{l=1}^{M} a_l D_{k-l+1}. \tag{3.5}$$

The coefficients a_l in (3.5) must be chosen in a way that minimizes the mean square error between $\hat{D}_k$ and D_k, i.e. $\mathbb{E}[(D_k - \hat{D}_k)^2]$. The desired coefficients are the solution of the set of the so-called normal equations [48]:

$$\sum_{m=0}^{M-1} a_{m+1} r_D(m-l) = r_D(l+1),\ l = 0, 1, \ldots, M-1. \tag{3.6}$$

In (3.6), $r_D = \mathbb{E}[D_k D_{k+l}]$ is the lag-l autocorrelation function of the process $\{D_k\}$. The exact form of the autocorrelation function is unknown, but it can be estimated using the past values of the process $\{D_k\}$. Suppose we have K values in the past, we can thus write

$$r_D(r) \simeq \frac{1}{K-|r|} \sum_{k=1}^{K-|r|} D_k D_{k+|r|}, \quad r = 0, \pm 1, \pm 2, \ldots, \pm(K-1). \tag{3.7}$$

The playout time of the i-th packet of talkspurt k is set as follows:

$$p_k^i = \begin{cases} t_k^1 + \hat{D}_k, & \text{for } i = 1 \\ p_k^1 + (t_k^i - t_k^1), & \text{for } 1 < i \leq N_k. \end{cases} \tag{3.8}$$

For very small values of p, there is a deviation on the overall perceived loss percentage from the one we desire. To deal with this deviation, for the range $0.005 \leq p \leq 0.02$, we allow our MA algorithm to slightly increase the playout delay by $\Delta_{\hat{D}_k} = f(p)\sqrt{\mathbb{E}[(\hat{D}_k - D_k)^2]}$, with $f(p) = -\delta \times (\frac{p}{p_{\max}} - 1)$, where δ is a constant controlling how much we increase the

playout delay as a function of the square root of $\mathbb{E}[(D_k - \hat{D}_k)^2]$. We set $p_{\max} = 0.02$ and $\delta = 0.5$. This allows to reduce considerably the deviation of the measured loss percentage from p, without impacting much the delay.

Bias and transformation

Our control on p is done by setting $\hat{D}_k$ to a value that minimizes the MSE between with the optimal playout delay per talkspurt. But the relationship between the playout delay and the loss percentage may not be linear. This may cause a deviation of the perceived loss percentage from the desired one.

To correct this bias we apply a transformation on D_k. Se we define $X_k = G(D_k)$. The prediction is done on the process X_k instead of D_k, using a Moving Average predictor, i.e., $\hat{X}_{k+1} = \sum_{l=1}^{M} a_l X_{k-l+1}$. Once the estimate of X_k, denoted by $\hat{X}_k$ is obtained, we set the playout delay to $G^{-1}(\hat{X}_k)$.

The function $G(x)$ must compensate for the non-linearity of the function $F(x)$. It must transform the error in setting the playout delay, so as to make $\hat{p}$ equal to p. Unfortunately, it is very difficult to find the expression of $G(x)$. Some approximations can be used. We give an example of a transformation that we use in this paper. Our measurements show that the function $F(x)$ is convex, and close to exponential. We consider then as transformation the exponential function, with a decay coefficient α, that is, we take $G(x) = e^{-\alpha x}$. Hence, we predict $X_k = e^{-\alpha D_k}$ instead of predicting D_k.

Hybrid algorithm

Based on our results (see [47]), we show that moving average estimation is an attractive approach for playout delay control. The two algorithms described above outperform Ramjee's algorithms on both loss percentage and average playout delay. Short traces combined with high network loss impact the performance of our algorithms since their accuracy depends on the number of talkspurts per session and on the number of packets per talkspurt.

A real online implementation is proposed as a hybrid algorithm. Our hybrid algorithm combines Ramjee's algorithm B and the moving average algorithm by applying the transformation described in the previous subsection. The idea is quite simple. During the first `MAXTKSP` talkspurts, Ramjee's algorithm B is executed, while at the same time samples of $\{D_k\}$ are collected and transformed on X_k as explained before. So, we collect enough information for starting the moving average (MA) estimation and we apply our MA algorithm with transformation starting from talkspurt `MAXTKSP + 1`. We call this algorithm "hybrid online algorithm", and we show that it performs well for the loss range of interest, and for most of the traces.

3.8.4 Conclusions

In this work we proposed three variants of a moving average algorithm for playout delay on VoIP. The strength of our scheme lies in the fact that we are able to tune the loss percentage p to a given desired value. This falls on relationship (2) of our QoS chain since controlling a measure of loss directly impacts the audio quality provided to a user by an audio application.

As the Internet is a best-effort network, providing end-to-end QoS is an important feature of any application, and in particular, real-time applications like VoIP.

Our algorithm predicts the optimal playout delay per-talkspurt, or a function of it, using the past history of the process. To reconstruct the periodic form of the stream of packets, the playout delay of packet in a talkspurt is based on the playout time of the first packet in the talkspurt. An interesting recent approach [49,50] shows that it is possible to adapt the playout delay at each packet arrival, leading to a better performance than in a talkspurt basis. Our future work will focus on per-packet playout delay adaptation.

The reader is referred to [47] for a detailed description of our work.

Chapter 4

Conclusions and Outlook

This document presented a state-of-the-art survey of user-perceived Quality of Service and quality feedback, which is the topic of the work package JRA.6.1 of the Network of Excellence "Euro-NGI". The state-of-the-art is presented as perceived and exemplified by the partners contributing to this work package. The whole chain from the user's perception of quality (including usability) via the application to network performance (monitoring and provisioning) needs to be covered, which is a promising basis for future joint research activities.

As indicated before, JRA.6.1 interfaces with many Euro-NGI workpackages. Within the activity on JRA.6 "Socio-Economic Aspects of Next Generation Internet", QoS from the user's point of view is related both to JRA.6.2 "Payment and cost models for Next Generation Internet" through taking economic incentives to users into account, and to JRA.6.3 "Creation of trust by advanced security concepts" by regarding security as a part of QoS.

Still, the link between user-perceived utility and network QoS has to be strengthened such that quality becomes even more quantifiable and thus better monitorable. This link has been studied intensively for audio and video traffic; however, there are many applications (or application components) left to study.

Furthermore, the (pseudo-) subjective ratings of user-perceived quality rely mostly on simulated network problems. Experiments with "real" network entities (e.g. in a controlled lab environment) may strengthen the link between user rating and typical network problems.

We have seen a considerable mismatch between standardization efforts and best practice in Internet, which can simply be summarized as "best effort". SLAs need to be established with simply-to-measure and easy-to-understand quality indicators unambigiously reflecting users' perception of service levels. This in turn implies the need for unambiguous terms. The one-stop service concept has to be established in the Internet context, which simplifies the user's life pretty much as he or she merely has one partner to deal with in case of trouble.

With regards to quality feedback, existing links have to be strengthened, while others need to be established, e.g.

- Feedback application (user side) → service/network provider for QoS-critical applications; monitor application-level performance parameters. Some kind of automatic feedback of this kind could be a dream scenario for all QoS-critical applications due to the fact that the user-perceived QoS can be used as input for quality control right away. However, such a solution might require a lot of effort. (The "manual version" of this feedback consists of user feedback (e.g. complains) to some kind of support or helpdesk.)
- Feedback application → user to be improved (example: some kind of "busy tone").

Also, verifying performance levels within a multi-service/technology/provider network by appropriate monitoring (which means finding a good compromize between precision and effort), developing appropriate control algorithms and management infrastructures are challenges for future work.

Glossary

3GPP	Third Generation Partnership Project
ATHENA	Auction-based THird gEneration Networks resource reservAtion
B2B	Business to Business
B2C	Business to Consumer
C2C	Consumer to Consumer
CDN	Content Delivery Network
CODIS	COntent Delivery Improvement by Satellite
DVB	Digital Video Broadcasting
eTOM	enhanced Telecom Operations Map (www.tmforum.org)
EURESCOM	European Institute for Research and Strategic Studies in Telecommunications (www.eurescom.de)
FCFS	First Come First Serve
GSM	Global System for Mobile communication
IBP	Internet Backbone Provider
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISO	International Standardization Organization
ISP	Internet Service Provider
ITU-R	International Telecommunication Union – Radiocommunication Sector
ITU-T	International Telecommunication Union – Telecommunication Standardisation Sector
JRA	Joint Research Activity
MMS	Multimedia Messaging Service
MOS	Mean Opinion Score
QoS	Quality of Service
RNN	Random Neural Network
SLA	Service Level Agreement
SMS	Short Messagaging Service
TCP	Transmission Control Protocol
TPM	Task oriented Performance Measure
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication System
VoIP	Voice over IP
WAP	Wireless Application Protocol
WLAN	Wireless Local Area network

Bibliography

[1] ITU-T. Recommendation E.860: Framework of a service level agreement, June 2002.

[2] A. Bouch, A. Kuchinsky, and N. Bhatti. Quality is in the eye of the beholder: Meeting user's requirements for internet quality of service. Technical Report HPL-2000-4, HP Laboratories Palo Alto, January 2000.

[3] ITU-T. Recommendation E.800: Terms and definitions related to quality of service and network performance including dependability, August 1994.

[4] ITU-T. Recommendation I.350: General aspects of quality of service and network performance in digital networks, including ISDNs. URL: http://www.itu.ch/.

[5] EQoS. EURESCOM Project P806. URL: http://www.fokus.gmd.de/research/cc/glone/projects/p806/content.html, last visited on 05/25/2004.

[6] ITU-T. Recommendation X.140: General quality of service parameters for communication via public data networks, September 1992. URL: http://www.itu.ch/.

[7] ITU-T. Recommendation X.642: Information technology - quality of service - guide to methods and mechanisms, September 1998. URL: http://www.itu.ch/.

[8] RFC 1958. Architectural principles of the Internet. URL: http://www.ietf.org/rfc/rfc1958.txt.

[9] H. Hlavacs, J. Lauterjung, G. Aschenbrenner, E. Hotop, and C. Trauner. QoS measurement. Deliverable D-3400, CODIS, November 2003.

[10] A. Bouch and M.A. Sasse. It ain't what you charge it's the way you do it: a user perspective of network QoS and pricing. In *Proceedings of IM'99, Boston MA*, 1999.

[11] H. Knoche, H.G. de Meer, and D. Kirsh. Utility curves: Mean opinion scores considered biased. In *Proceedings of IWQoS'99*, 1999. http://www.cs.ucl.ac.uk/research/iwqos99/papers/.

[12] ITU-T. Recommendation P.910: Subjective video quality assessment methods for multimedia applications, September 1999. URL: http://www.itu.ch/.

[13] ITU-R. Recommendation BT.500-10: Methodology for the subjective assessment of the quality of television pictures., March 2000. URL: http://www.itu.ch/.

[14] J. Nielsen. *Usability Engineering*. Morgan Kaufman, San Francisco, 1994.

[15] R. Rajamony and M. Elnozahy. Measuring client-perceived response times on the www. In *3rd USENIX Symposium on Internet Technologies and Systems (USITS)*, San francisco, March 2001.

[16] Peakstone. eAssurance Concept Guide, 2001. URL: http://www.peakstone.com/.

[17] N. Bhatti and R. Friedrich. Web server support for tiered services. Technical report, HP Laboratories Palo Alto, 1999.

[18] Zona Research Inc. The economic impacts of unacceptable web-site download speeds, April 1999.

[19] A. Bouch, M.A. Sasse, and H. de Meer. Of packets and people: A user-centered approach to Quality of Service. In *Proceedings of IWQoS'00*, 2000.

[20] N. Bhatti, A. Bouch, and A. Kuchinsky. Integrating user-perceived quality into web server design. In *Proceedings of WWW'00*, Amsterdam, 2000.

[21] RFC 3550. RTP: A transport protocol for real-time applications. URL: http://www.ietf.org/rfc/rfc3550.txt.

[22] T. O'Neil. Network-based quality of service for IP video conferencing. White paper available at http://www.polycom.com/, Polycom Inc., 2002.

[23] Scandinavian Airlines. http://www.sas.se.

[24] D.D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in an integrated services packet network: Architecture and mechanism. In *Proceedings of SIGCOMM'92*, pages 14–26, 1992.

[25] Inc. Decisys. Service level agreements: Demonstrating how your network meets business objectives and user expectations. Novel Users International. Last viewed May 12th, 2004.

[26] D. Chalmers and M. Sloman. QoS and context awareness for mobile computing. *Lecture Notes in Computer Science*, 1707:380ff, 1999.

[27] N. Bhatti, A. Bouch, and A. Kuchinsky. Integrating user-perceived quality into Web server design. *Computer Networks (Amsterdam, Netherlands: 1999)*, 33(1–6):1–16, 2000.

[28] P. Coverdale. Multimedia qos requirements from a user perspective. In *ITU-T Study Group 12 Workshop on QoS and user-perceived transmission quality in evolving networks*, Senegal, 10 2001.

[29] Jupiter II. EURESCOM Project P807-GI. URL: http://www.eurescom.de/~public-webspace/P800-series/P807/index.html.

[30] A. Bouch. Quality is in the eye of the beholder: Meeting users's requirements for internet quality of service. In *Proceedings of ACM conference on Human Factos in Computing Systems (CHI 2000)*, The Hague, The Netherlands, 2000. ACM.

[31] A. Kassler, T. Guenkova-Luy, D. Mandato, P. Schoo, I. Armuelles, T. Robles, P. Ruiz, and A. Bascuana. Enabling mobile heterogeneous networking environments with end-to-end user perceived QoS - the BRAIN vision and the MIND approach. *EWC2002*, 02 2002.

[32] P. M. Ruiz. Managing the user-perceived QoS in heterogeneous IPv6 networks. *Global IPv6 Summit*, 2003.

[33] A. Ferscha, J. Johnson, G. Kotsis, and C. Anglano. Pro-active performance management of distributed applications. In A. Boukerche, S. Das, P. Wilsey, and C. Williamson, editors, *Proceedings of the Sixth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS'98*, pages 146–152. IEEE Computer Society Press, 1998. ISBN 0-8186-8566-2.

[34] G. Kotsis. Performance management in ubiquitous computing environments. In S.V. Raghavan and S.P. Mudur, editors, *Proceedings of the 15th International Conference on Computer Communication (ICCC 2002)*, pages 988–997. ICCC Press, 2002. ISBN 1-891365-08-8 (invited paper).

[35] T. Grill, K. I. Ibrahim, and G. Kotsis. Agents for ambient intelligence — support or nuisance. *OeGAI Journal*, 23:19–26, 01 2004.

[36] H. Hlavacs, G. Aschenbrenner, E. Hotop, A. Baijal, and A. Garg. QoS analysis method. Deliverable D-2300, CODIS, July 2002.

[37] ETSI. Digital Video Broadcasting (DVB) measurement guidelines for DVB systems. Standard 101 290 V1.2.1, ETSI, May 2001.

[38] S. Mohamed and G. Rubino. A study of real-time packet video quality using random neural networks. *IEEE Transactions On Circuits and Systems for Video Technology*, 12(12), December 2002.

[39] S. Mohamed, G. Rubino, and M. Varela. Performance evaluation of real-time speech through a packet network: a random neural networks-based approach. *Performance Evaluation Journal*, 57(2):141–161, 2003.

[40] G. Rubino and M. Varela. A new appraoch for the prediction of end-to-end performance of multimedia streams. In *QEST: Quantitative Evaluation of SysTems, 2004*, Twente, September 2004.

[41] M. Fiedler and K. Tutschku. Application of the Stochastic Fluid Flow Model for bottleneck identification and classification. In *Proceedings of 2003 Design, Analysis, and Simulation of Distributed Systems (DASD 2003)*, pages 35–42, Orlando, USA, April 2003.

[42] M. Fiedler, K. Tutschku, P. Carlsson, and A.A. Nilsson. Identification of performance degradation in IP networks using throughput statistics. In J. Charzinski, R. Lehnert, and P. Tran Gia, editors, *Providing Quality of Service in Heterogeneous Environments. Proceedings of the 18th International Teletraffic Congress (ITC-18)*, pages 399–407, Berlin, Germany, September 2003.

[43] The 3rd Generation Partnership Project (3GPP). URL: http://www.3gpp.org/.

[44] C. Bettstetter, H.-J. Vogel, and J. Eberspächer. GSM phase 2+, general packet radio service GPRS: Architecture, protocols and air interface. *IEEE Communications Surveys*, 2(3), 1999.

[45] T. Groves and J. Ledyard. Optimal allocation of public goods: A solution to the 'free rider' problem. *Econometrica*, 45(4):85–96, 1997.

[46] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne. Adaptive playout mechanisms for packetized audio applications in wide-area networks. In *Proceedings of the IEEE Infocom*, pages 680–688, 1994.

[47] V. Ramos, C. Barakat, and E. Altman. A moving average predictor for playout delay control in VoIP. In *Proceedings of the XI International Workshop on Quality of Service (IWQoS)*, pages 155–173, 2003.

[48] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, algorithms, and applications*. Prentice-Hall Inc., 1996.

[49] Y.J. Liang, N. Farber, and B. Girod. Adaptive playout scheduling and loss concealment for voice communications over IP networks. *IEEE Transactions on Multimedia*, April 2001.

[50] Y.J. Liang, N. Farber, and B. Girod. Adaptive playout scheduling using time-scale modification in packet voice communications. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing ICASSP*, volume 3, pages 1445–1448, May 2001.