Quality Management in Multimedia Applications

Thomas R. Schmidt <schmidt@ind.uni-stuttgart.de>
University of Stuttgart, Institute of Communication Networks and
Computer Engineering, Pfaffenwaldring 47, D-70569 Stuttgart, Germany

Abstract

Multimedia communication has become a service that is used more and more frequently. Users now expect to receive a certain end-to-end Quality of Service (QoS) at their terminals. Therefore, the terminal has to implement a Quality of Service control mechanism that tries to optimize the parameter setting. In order to achieve this, the *experienced QoS* at the user is measured and then compared to the *expected QoS* for the specific type of application.

If the experienced and the expected QoS differ, the application has to adjust its parameters. To do this, quantitative knowledge of the influence of certain parameter settings is necessary. At the University of Stuttgart, an `eXperimentation Platform` (`XP`) is being developed that allows the investigation of the influence of parameters in the terminals as well as the influence of and on the network infrastructure for multimedia communication applications.

1 Introduction

In a multimedia communication between two or more users at terminals connected via a network, the users expect a certain Quality of Service (QoS) provided by the end-to-end communication system at the terminals. Due to resource limitation, it is not possible to just set the parameters of the media transmissions to the best values. Instead it is necessary to tune the parameters against each other in order to reach the best QoS given the available resources. Depending on the type of application (e.g. video conferencing, Internet telephony), a specific set of values (like video picture size or coding formats) should be used to achieve the expected perceived application performance. One way would be to let the user enter these parameters – but this would require an enormous amount of knowledge on his part and an adaptation to changing resource availiability would be very difficult to achieve. Therefore an automated quality management mechanism is required.

During the communication, the quality management can measure (or receive from other terminals) parameters that describe the actual QoS experienced by the users. It compares the resulting with the expected QoS and can then control the mapping of the expected quality onto the tunable parameters when necessary, in order to improve the matching of the experienced with the expected QoS.

One influence on this mapping is the user who can redefine his expectations in view of the quality the application can provide. On the other hand, the network could provide constraints to the mapping function, influencing the traffic the application generates. This could be very useful in networks where the access mechanism does not provide a means of access control.

The goal of our work in progress is to define, design and implement an experimentation platform (`XP`) that allows the investigation of the quality management strategies. Based on the acquired data, simulations can be used to discover and study quantitative dependencies between parameter values and experienced QoS.

In the second section, an overview of some of the currently used multimedia systems will be given, pointing out interesting approaches. The third section will describe the ideas behind our approach, the goals and the methods we are using. The current status of the `XP` is given in the forth section. Finally a conclusion and a description of the planned research work are given in the fifth section, also highlighting the potential of this design.

2 Background and motivation

Multimedia communication applications are becoming more and more popular. One of the most widely used is “NetMeeting” by Microsoft [Mic99]. While this application may be adequate for the immediate usage and low quality applications, it does not allow the user to observe the internal mechanisms it uses or even influence its behaviour.

Another system for multimedia communications is the set of *MBone* tools [Kum95]: programs like *vat* or *rat*, *vic*, *sdr* [Fah97] are combined to allow (multicast) sessions to take place between two or many more users. As these programs are available as source code, many modifications and extensions have been developed that try to tackle certain shortcomings of the original versions: the lack of a tool-overlapping resource control mechanism, the complexity of session setups and of course the usability with specific hardware components.

There is also a variety of proprietary systems that often enclose specialized hardware like video encoding boards (e.g. *CellStack Capella Video* [Cel99], native ATM). In most cases, these applications are not very flexible in terms of bandwidth use (e.g. *PictureTel* [Pic98], n×64 Kb/s ISDN) or adaptability on the current transmission conditions (e.g. in the Internet). Though they provide a good quality, their high cost will prohibit a widespread use of these systems.

Of course, many research systems have been developed (an overview can be found in [CAH96], e.g.). Most of these focus on the terminal, either trying to allow the user to easily construct an application tailored to his needs (e.g. *CINEMA* [BDHea93]), or investigating the local resource management in the end systems (e.g. *QoS Broker* [NS95]).

The focus of our research lies in the effects of multimedia communication on a transport network and how it is influenced by it. But currently, comparative studies can only be made by comparing completely different applications. Keeping in mind the enormous number of parameters that influence the QoS of these services, this does not allow to identify and quantify the effects of a specific approach. For example, systems using native ATM do not run on a LAN environment, making direct comparisons impossible.

Therefore the `eXperimentation Platform` (`XP`) has been designed. It allows to study the effects of a single modification to the setup of an application, without altering other parameters in the setup. E.g., it is possible to completely replace the adaptation mechanism without changing codecs, signalling or network modules; or to change only the network module from native ATM to IP/RSVP.

3 Design

3.1 Basic principle

The central idea of the `XP` design is the separation of the QoS-management and the media transmission functionality (fig. 1). This allows the exchange of the QoS mapping and adaptation mechanisms in order to investigate their effects, depending on factors like media codecs and network access.

In the QoS-management layer, a concatenation of several QoS mapping steps (fig. 2) is performed. In the first step, the user enters a *User Level Quality* (ULQ), that defines the quality of a service as a whole. In order to relieve the user of the burden to specify technical parameters like bit rate or delay, this parameter is given as a number between 0 and 10. So, as an example, the user can demand a very good quality telephony application by specifying a ULQ of 9.

It is now the task of the `XP` to convert this figure into an *Application Level Quality* (ALQ). Since every application has its own range of QoS parameters, the mapping of the ULQ on factors like the bit rate or the delay must be done differently. E.g., a telephony application at ULQ 9 may wish a very low delay, while in a tele teaching scenario, the delay may be very high for the non-interactive case.

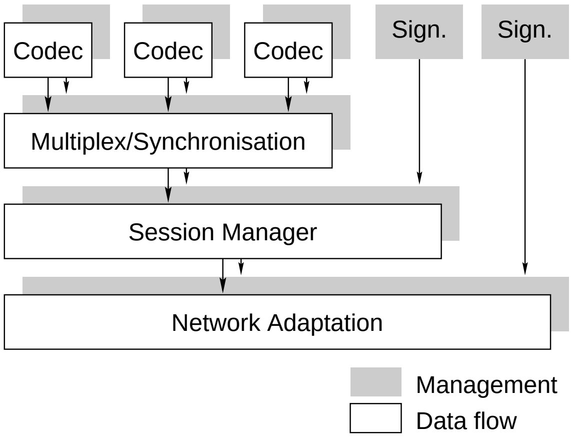

Figure 1: System layers, cf. fig. 4

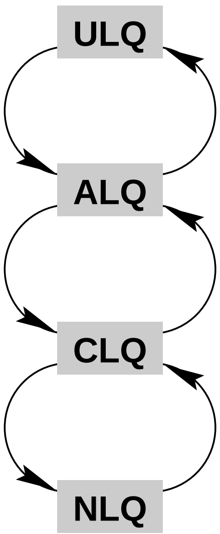

Figure 2: cascaded QoS levels

After the XP has determined the media parameters that define the QoS of the service it should deliver, it choses the means to do so; that is it selects codecs and parameters for them and assigns system resources (fig. 3). Once the transmission starts, the traffic produced will have a certain characteristic, specified as the *Communication Level Quality* (CLQ).

Depending on the type of transport service used, the *Network Level Quality* (NLQ) denotes the parameters that are expected of this service in order to grant the CLQ. The transmission will experience a certain NLQ on its way to the receiver, where the quality of the decoded media will be represented in the parameters on each level, this time in the reverse direction.

In the next step the *experienced* QoS is compared to the expected QoS and an adaptation mechanism is used to change the parameters in the sender and the receiver in order to improve the match between both (fig. 3).

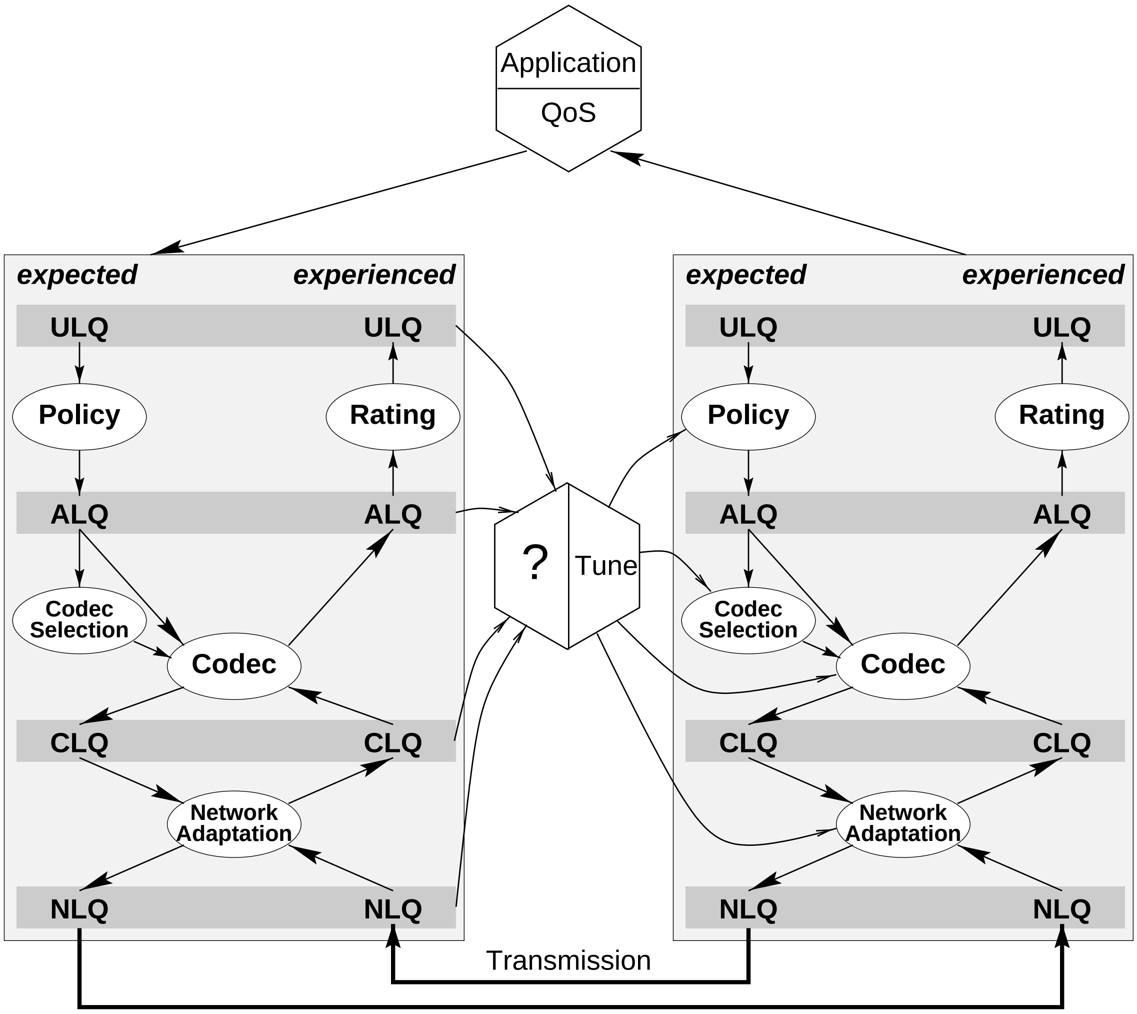

Figure 3: Parameter mapping and adaptation cycle

Example. The system is set up as an *Internet Telephony* application. The user *expects* an ULQ of 8, i.e. good quality. Now the *policy* choses the ALQ values for this combination, e.g. an end-to-end delay of up to 100 ms and a very good sound quality (8 kHz, 8 bit μ-law, i.e. 64 Kb/s). The appropriate codecs are used and provided with parameters, which results in a specific traffic, described in the CLQ (a 320 byte frame every 40 ms). Depending on the network used to transport the data, a NLQ results, e.g. for RTP/UDP a 355 byte packet every 40 ms, i.e. 71 Kb/s.

The packets arrive with a certain delay and loss probability at the receiver, described in the NLQ (adding a delay jitter value). At the CLQ level, packets that are too late are discarded and therefore the codec will not play back exactly the same audio signal, which is described in the ALQ (e.g. the delay here is 60 ms but the sound quality is not good due to the losses). The *rating* function can now determine the *experienced* ULQ as being 6.

Using signalling, the experienced QoS-values are transmitted to the sender who now tries to *tune* its parameter setting in order to improve the experienced QoS at the receiver. In the example, this could mean to change the codec to ADPCM, reducing the total amount of data transmitted to 39 Kb/s but also resulting in a higher total delay of 90 ms.

3.2 Layout

The XP has been designed as a highly modular system where modules can be exchanged, added and modified easily without changing other parts. Therefore the influence of a specific modification can be investigated more easily, since the rest of the system will stay the same.

The system has a fixed basic structure (fig. 4) built out of *abstract modules* that are implemented by *specialized modules*. These modules can be taken out of a library of modules, where a module can combine the tasks of several other modules, e.g. a MPEG module could combine audio and video encoding, synchronization and multiplexing.

Between the modules, it is possible to insert “probe points”: modules that can monitor the traffic generated by the modules it sits between. The generated traces can be used to model the characteristics of a certain service or codec. They also allow to compare the characteristics of the traffic generated by the sender with that reaching the receiver after it has been transmitted over the network. Together with the evaluation of the experienced QoS it is possible to evaluate the effects the transmission has on the overall QoS. This allows the design of a useful adaptation mechanism to compensate for the modification the network inflicts on the traffic.

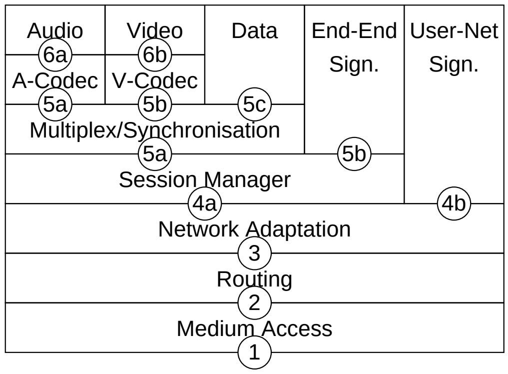

Figure 4: Architecture with probe points

4 Status

The system has been implemented as a working prototype. Currently, the focus lies on the realization of the standard continuous media types for audio and video, i.e. audio codecs like *G.711* and *GSM* as well as video codecs like *Motion-JPEG* and *H.261*. Further work is put into the QoS-adaptation mechanism and the signalling. Ultimately, the system will support several protocols, including *H.323*.

5 Conclusion and further work

The use of multimedia communication services has increased dramatically in the last years. More and more it became clear that in order to provide a sufficient end-to-end QoS, the management of the network *and* the terminal resources must be coordinated.

Many approaches to reach this goal have been invented. Unfortunately, it is very difficult to compare the advantages and disadvantages of these approaches, since most of the time the specific algorithms are either directly interweaved with an application or are the only modular parts in the setup.

This was the reason for the design of our `eXperimentation Platform`, a system that seperates functions into modules that can be assembled to form a multimedia application. With this application it is possible to investigate many different combinations of network modules, QoS management algorithms, media codecs etc. in order to evaluate their effects on the network and also their ability to cope with network-inflicted changes of the transported traffic. The observation of the correlation between the parameters and the experienced QoS will allow the design of better adaptation and resource management algorithms.

Currently, the following studies are planned:

- interaction of terminal QoS-adaptation algorithms with network resource reservation mechanisms, especially *RSVP-over-ATM* ([BZBea97]; implemented in an European project)
- effects of data stream prioritizing on the overall QoS and the network, e.g. by using *Differentiated Services* [BBCea98]
- effects of the usage of secure data transmission (i.e. encryption) on the QoS and the signalling
- evaluation of user-experienced QoS by interviewing a non-technical audience
- usage of the results for modelling the traffic-generation, -transmission and the experienced QoS in order to investigate basic principles (This will be supported by a simulation library already implemented at the institute.)

Even after these studies are finished, the `XP` can be used as a basis for further work to evaluate new network protocols, to implement new applications and media codings and to deepen the understanding of the mechanisms to provide QoS in multimedia applications.

References

[BBCea98] Blake, S.; Black, D.; Carlson, M.; Davies, E.; Wang, Z.; Weiss, W.: *An Architecture for Differentiated Services, RFC 2475*, IETF, December 1998.

[BDHea93] Barth, I.; Dermler, G.; Helbig, T.; Rothermel, K.; Sembach, F.; Wahl, T.: *CINEMA: Eine konfigurierbare, integrierte Multimedia-Architektur*, GI/ITG Arbeitstreffen 'Verteilte Multimedia Systeme', Stuttgart, D, 18.02.1993 - 19.02.1993, Band 5, K.G. Saur, Februar 1993, ISBN 3-598-22407-9, S. 33–47.

[BZBea97] Braden, R.; Zhang, L.; Berson, S.; Herzog, S.; Jamin, S.: *Resource ReSerVation Protocol (RSVP)*, RFC 2205, IETF, September 1997.

[CAH96] Campbell, A.; Aurrecoechea, C.; Hauw, L.: *A Review of QoS Architectures*, ftp://ftp.ctr.columbia.edu/CTR-Research/comet/public/papers/96/CAM96a.ps.gz, March 1996.

[Cel99] *CellStack Systems*, http://www.cellstack.com, 1999.

[Fah97] Fahner, H.: *MBone Software Archive*, http://www-ks.rus.uni-stuttgart.de/mbone/mb-soft.html, 1997.

[Kum95] Kumar, V.: *MBone: Interactive Multimedia on the Internet*, New Riders, Indianapolis, USA, 1995, ISBN 1-56205-397-3.

[Mic99] *Microsoft NetMeeting*, http://www.microsoft.com/netmeeting/, January 1999.

[NS95] Nahrstedt, K.; Smith, J. M.: *The QOS Broker*, IEEE Multimedia, Vol. 2, No. 1, USA, Spring 1995, pp. 53–67.

[Pic98] *PictureTel Corporation*, http://www.picturetel.com, 1998.