Project Number:	**IST-1999-10077**
Project Title:	**AQUILA** **Adaptive Resource Control for QoS Using an IP-based Layered Architecture**
Deliverable Type:	**PU - public**

Deliverable Number:	**IST-1999-10077-WP2.2-TUD-2202-PU-O/b0**
Contractual Date of Delivery to the CEC:	**March 31, 2001**
Actual Date of Delivery to the CEC:	**March 29, 2001**
Title of Deliverable:	**Description of user applications for the first trial**
Workpackage contributing to the Deliverable:	**WP 2.2**
Nature of the Deliverable:	**O – Other (Specification)**

Editor:	**Falk Fünfstück (TUD)**
Author(s):	**Tina Catsora, Andreas König, Manfred Leisenberg, Sebastian Peitsch, Andreas Pollpeter, Jochen Röndigs (BAG), John Karadimas (QSY), Sotiris Maniatis, Haris Tsetsekas (NTU), Falk Fünfstück, Anne Thomas (TUD)**

Abstract:	This deliverable D2202 describes the functionality, namely the user applications, offered to the end-users of the first trial. Legacy applications are the focus of the first trial. This deliverable depicts their requirements and how they are supported by the EAT.

Keyword List:	AQUILA, IST, Legacy Application, Requirements, QoS

Executive Summary

This deliverable describes the application functionality offered to the end-users of the first trial. Existing multimedia offers and applications will be the focus of the first trial. Therefore, the deliverable concentrates on so-called legacy applications and how to support them.

Although a full integration with the End-user Application Toolkit (EAT) is not yet expected for the first trial but for the second one, the applications mentioned in this deliverable will benefit from a skeleton EAT which is already specified in [D2201]. In detail, the deliverable describes how applications will be integrated in the AQUILA approach for the first trial.

Table of Contents

1 INTRODUCTION 8

2 ANALYSIS OF EXISTING MULTIMEDIA OFFERS 10

2.1 INTRODUCTION – DEFINITION OF MULTIMEDIA INTERNET SERVICES 10

2.2 ONLINE GAMES 10

2.2.1 General 10

2.2.2 Online Games System Requirements 11

2.2.3 Quality Issues and Online Games 11

2.3 VIDEOCONFERENCING 12

2.3.1 General 12

2.3.2 Videoconferencing System Requirements 13

2.3.3 Quality Issues and Videoconferencing 13

2.4 VOICE OVER IP 14

2.4.1 General 14

2.4.2 Voice over IP System Requirements 16

2.4.3 Quality Issues and Voice over IP 16

2.5 STREAMING MEDIA 16

2.5.1 On2.com 19

2.5.1.1 On2.com General Information 19

2.5.1.2 On2.com System Requirements 20

2.5.2 Big Brother 20

2.5.2.1 Big Brother General Information 20

2.5.2.2 Big Brother Online System Requirements 21

2.5.3 Bertelsmann Broadband Group 21

2.5.3.1 Broadband Group General Information 21

2.5.3.2 Bertelsmann Broadband Group System Requirements 22

2.5.4 Music Mediazine 22

2.5.4.1 Music Mediazine General Information ... 22
2.5.4.2 Music Mediazine System Requirements ... 23
2.5.5 Quality Problems on Streaming Media Reported by the End-user and Explanation Given by the Provider 23
2.5.6 General QoS Related Problems on Streaming Media ... 24
2.6 SIGNIFICANT FACTORS FOR MULTIMEDIA INTERNET SERVICES ... 25
2.7 NEW PRODUCTS ... 27
2.7.1 RealProducer 8 Plus ... 27
2.7.2 On2.com True Motion VP 3.2 ... 28
3 REQUIREMENTS OF LEGACY APPLICATIONS ... 29
3.1 STREAMING MEDIA: REALSYSTEM ... 30
3.1.1 Description ... 30
3.1.2 Configuration ... 31
3.1.3 Technical requirements ... 33
3.2 MULTIMEDIA CONFERENCING: NETMEETING ... 34
3.2.1 Introduction ... 34
3.2.2 Configuration ... 34
3.2.3 NetMeeting bandwidth considerations ... 36
3.2.4 Technical requirements ... 37
3.2.5 Session characteristics ... 38
3.3 IP TELEPHONY: WINSIP ... 41
3.3.1 Description ... 41
3.3.1.1 WinSip ... 41
3.3.1.2 Significance of VoIP applications ... 43
3.3.1.3 Applications implementing IP Telephony ... 44
3.3.1.4 VoIP applications and quality ... 44
3.3.2 Configuration and usage ... 46
3.3.2.1 Session configuration ... 46

3.3.2.2 User guide 47
3.3.3 Technical requirements 49
3.4 MULTI-PLAYER ONLINE GAMES: UNREAL TOURNAMENT 49
3.4.1 Description 49
3.4.2 Significance of multi-player online games 50
3.4.3 Configuration and usage 51
3.4.3.1 Session configuration 51
3.4.3.2 User Guide 52
3.4.4 Technical requirements client 57
3.4.5 Technical requirements server 58
3.5 MULTI-PLAYER ONLINE GAMES: ULTIMA ONLINE 59
3.5.1 Description 59
3.5.2 Connection problems 60
3.5.3 Technical requirements 62
4 APPLICATION PROFILES FOR LEGACY APPLICATIONS 64
4.0 APPROACH: REFINEMENT OF [D2201, CHAP. 4.2] 64
4.0.1 Structure of the application profile 64
4.0.2 Analysis patterns in the application profile structure 65
4.0.3 Realisation of the profiles 66
4.0.3.1 Solution proposal 67
4.0.4 Refined application profile as XML / DTD 70
4.1 PROFILE FOR REALSYSTEM 73
4.2 PROFILE FOR NETMEETING 76
4.3 PROFILE FOR WINSIP 79
4.4 PROFILE FOR UNREAL TOURNAMENT 80
4.5 PROFILE FOR ULTIMA ONLINE 82
5 SCENARIOS FOR THE FIRST TRIAL 83
5.1 APPLICATION LEVEL QOS TESTS/VALIDATION SCENARIOS 83

5.1.1 RealSystem ... 84
5.1.2 NetMeeting ... 84
5.1.3 WinSip ... 85
5.1.4 Unreal Tournament ... 85
5.1.5 Ultima Online ... 85
5.2 NETWORK LEVEL QOS TESTS/VALIDATION SCENARIOS ... 85
5.2.1 RealSystem ... 86
5.2.2 NetMeeting ... 86
5.2.3 WinSip ... 86
5.2.4 Unreal Tournament ... 86
5.2.5 Ultima Online ... 87
6 CONCLUSIONS FOR THE EAT AND THE SECOND TRIAL ... 88
7 ABBREVIATIONS ... 90
8 REFERENCES (ALL) ... 94

Table of Figures

FIGURE 2-1: SCHEMATIC REPRESENTATION OF THE VIDEOCONFERENCING TECHNOLOGY ... 12
FIGURE 2-2: SCHEMATIC REPRESENTATION OF THE VOICE OVER IP TECHNOLOGY ... 15
FIGURE 2-3: SCHEMATIC REPRESENTATION OF THE STREAMING TECHNOLOGY ... 18
FIGURE 3-1: SCREENSHOT OF REALPLAYER ... 31
FIGURE 3-2: SCREENSHOT OF REALPLAYER PREFERENCES ... 32
FIGURE 3-3: SCREENSHOT OF NETMEETING ... 34
FIGURE 3-4: NETMEETING CALL SET UP PROCEDURE ... 38
FIGURE 3-5: SCREENSHOT OF WINSIP ... 41
FIGURE 3-6: SCREENSHOTS OF THE DIFFERENT COMPONENTS OF A SESSION DESCRIPTION ... 43
FIGURE 3-7: CONFIGURATION DIALOG OF THE PREFERENCES FOR A SESSION ... 46
FIGURE 3-8: SESSION CONFIGURATION: MEDIA AND CONTACT ... 47
FIGURE 3-9: BUDDY DATA: SIP ADDRESS ... 48
FIGURE 3-10: SCREENSHOT OF THE NETWORK CONFIGURATION ... 52
FIGURE 3-11: SCREENSHOT OF THE SERVER ... 53
FIGURE 3-12: SCREENSHOT OF THE GAME SERVER SETTINGS ... 54

FIGURE 3-13: SCREENSHOT OF ULTIMA ONLINE 60
FIGURE 4-1: CLASS DIAGRAM OF THE APPLICATION PROFILE STRUCTURE 65
FIGURE 4-2: APPLICATION PROFILE CLASS DIAGRAM. THE ADAPTED ITEM-ITEM DESCRIPTION ANALYSIS PATTERN AND THE ADAPTED ACCOUNTABILITY KNOWLEDGE LEVEL ANALYSIS PATTERN ARE DEPICTED USING UML.. 66
FIGURE 4-3: JAVA WEB APPLICATION AND THE DIFFERENT TECHNOLOGIES 68
FIGURE 4-4: ARCHITECTURE PROPOSAL FOR THE EAT 69
FIGURE 4-5: RELATIONS BETWEEN THE DIFFERENT ELEMENTS, AND "CALLING" PROPOSAL 69
FIGURE 4-6: APPLICATION PROFILE GRAPH 70
FIGURE 4-7: APPLICATION PROFILE DTD FILE 72
FIGURE 4-8: APPLICATION PROFILE XML FILE 73
FIGURE 4-9: REALSYSTEM XML PROFILE WITH SERVICEID 76
FIGURE 4-10: NETMEETING XML PROFILE WITH SERVICEID 79
FIGURE 4-11: WINSIP XML PROFILE WITH BOTH SERVICEID AND QOSSPEC 80
FIGURE 4-12: UNREAL TOURNAMENT XML PROFILE WITH BOTH SERVICEID AND QOSSPEC 82
FIGURE 4-13: ULTIMA ONLINE XML PROFILE WITH BOTH SERVICEID AND QOSSPEC 82

Table of Tables

TABLE 2-1: GENERAL DESCRIPTION OF AVAILABLE CONNECTION MODES 26
TABLE 3-1: REALSYSTEM REQUIREMENTS 33
TABLE 3-2: NETMEETING BANDWIDTH TESTS 37
TABLE 3-3: NETMEETING REQUIREMENTS 40
TABLE 3-4: WINSIP APPLICATION REQUIREMENTS 49
TABLE 3-5: UNREAL TOURNAMENT CLIENT REQUIREMENTS 57
TABLE 3-6: UNREAL TOURNAMENT SERVER REQUIREMENTS 58
TABLE 3-7: ULTIMA ONLINE REQUIREMENTS 62

1 Introduction

This document presents the applications of the first AQUILA trial. All the 1st trial applications are legacy applications. They are not QoS-aware but can *indirectly* benefit from the QoS capabilities of the Resource Control Layer. That means, the applications are used as they are, i.e. without any modifications, in order to show that AQUILA can offer them QoS support.

The architecture to support this kind of applications is described in detail in [D1201, and D2201].

The *multimedia* Internet applications/services that may benefit from the AQUILA approach are categorised into four groups:

- Streaming Media,
- Multimedia Conferencing (Videoconferencing),
- IP Telephony (Voice over IP), and
- Multi-player Online Games.

The chapter 2 gives some general information and typical usage scenarios for these application categories. The general (network) requirements and quality issues are depicted. The aim of this chapter is, moreover, to give an overview of the state-of-the-art of multimedia applications and services. This is important in order to identify suitable QoS scenarios and to show, how *real* applications can benefit from QoS.

In the chapter 3, at least one well known and frequently used, for its category typical application is described in more detail. These are:

- RealSystem (RealPlayer/Server) for Streaming Media,
- Microsoft NetMeeting for Multimedia Conferencing,
- WinSip for IP Telephony (Voice over IP), and
- Unreal Tournament as well as Ultima Online for Multi-player Online Games.

A description of the functionalities, some screenshots, configuration possibilities (mainly concerning network and quality settings), and technical requirements constitute the main part of this chapter. The technical requirements are summarised in consistent tables which is the basis for the next step: the specification of the application profiles.

The chapter 4 starts with an introduction of the application profile approach. Application profiles are the representation way of at least legacy applications, in order to be supported by the

EAT. The chapter 4.0 is a refinement of [D2201, chapter 4.2]. Next, the technical and session characteristics of each selected application of the 1st trial are specified in corresponding XML profiles. The purpose is a twofold: First, the technical characteristics of the profiles can be used for manual, advanced QoS reservations via the EAT. Second, the profiles are needed for the preparation of the EAT's Legacy Application GUI that allows the reservation request on a much more abstract level.

The chapter 5 shortly depicts possible application and QoS validation scenarios. This is done for each 1st trial application and at two levels: network as well as application level. Another purpose is to show the difference between the possible 1st trial scenarios and the real world.

The last chapter 6 summarises the application requirements for the further development of the EAT, and make some proposals for the second trial applications.

2 Analysis of Existing Multimedia Offers

2.1 Introduction – Definition of Multimedia Internet Services

During the last years the world is getting more and more digitised. The communication and entertainment industries have been improved enormously, as they are influenced strongly by the wide spread and development of the Internet.

There is a big number of applications, entering our everyday lives and eventually becoming a part of them such as multimedia streaming services, videoconferences, online games and communications via the Internet. All these applications are based on the Internet protocol (IP). This term IP-based-services incorporates all these services using the latest Internet technology for transmitting media through networks, promising to lower the physical boundaries of the world. The more these applications are getting popular, the bigger the demand for raising the quality of these services.

In the following sections the most important Multimedia Internet Services will be described and analysed in brief as well as representative examples of them will be presented. Particular emphasis will be given on the presentation of streaming media and the quality issues involved.

2.2 Online Games

2.2.1 General

"Online games" is an application area of the Multimedia Internet Services which becomes more and more popular. There is already a big market dealing with the development of online games (e.g. the new game "DIABLO II" had sold 1.000.000 copies within the first two weeks after it was launched) and a big number of users that had integrated them into their everyday life.

Generally speaking, a **Multi-player Online Game** is a computer game where two or more players play together or against each other via a network (LAN, WAN (Internet), etc.). Nearly all of the online games include network functionality; they employ servers, newsgroups and game sites all over the Internet.

In simple words, the basic technology employed in the majority of online games is like follows: The system involved is almost every time based on a *client-server architecture*. There is the need of the operation of one server where the users (e.g. clients) are connected to, in order to play together. The clients are standard multimedia stations, which have installed the special software for the particular game. When the client starts the game on its local machine, it sends its settings over the network to the server. The server is handling these basic settings and calculates the most of the interaction data between the clients. The server is mainly responsible for distributing the data of the game over the network, as fast as possible.

Therefore, according to the basics of the online games technology, the software of the game is installed in the local workstations of the users, as well as the server. The settings data of each of the users are transferred (streamed) to the server, as well as the calculations of these settings are send (streamed) back to the clients.

2.2.2 Online Games System Requirements

There is no way to define in general, minimum system requirements for online games, as these depend very much on the game itself. On the side of the provider, the server should be a powerful machine, with extensive RAM space, one or two fast CPUs, fast processing hard disk (HD) and an Internet connection. On the side of the clients, a multimedia station (e.g. Pentium III 600, 128 MB RAM, 32 MB 3D Graphics Card, fast processing HD, high performance soundcard) is recommended.

2.2.3 Quality Issues and Online Games

Quality is an important factor for obtaining good results and popularity in the field of online games. These games have to be fun in order to be accepted by the end-user. If quality is poor, then the game is not so fun therefore it is not able to create a good business. Poor quality can be introduced because of:

- The network (e.g. when there is no real time transfer of the gaming data or the bandwidth is too low). Poor quality is responsible for causing problems such as latency (delays on transferring settings and control data) and packet loss (a number of these data is lost while transmitted through the network):
  - By the term of "latency" is meant whatever effect as the result of the delays the gaming data are transferred over the network. When this delay is over 120 ms it is nearly impossible to continue playing the online game, as such a big lag on receiving the data game information can be rather critical for the remaining course of the game.
  - The term "packet loss" implies that information might be lost between the client and the server. This is rather important, as then it is no possible to continue playing the game. When an action is send to the server and the server is not able to receive it, then this action counts like it was never send.
- The user's station (when it has not enough computational power). This causes problems on the client/server performance. If the CPU of the user is fully used for gaming computing there is no processor time for network computing, i.e. sending the game settings to the server (latency and packet loss). If the client or the server has performance problems this might lead to a network problem (latency).
- The bandwidth of the network. When the bandwidth is low, the transfer of the game settings (numbers, vectors, etc. but NO graphics or video streaming) have delays or they are not send at all. The majority of online games come along with ISDN connection, for ensuring quality connection to the network.

The solution of ensuring high quality on online games is the resource reservation between the server and the client. A secondary issue is the increase of the available bandwidth in order to improve the network speed connection.

2.3 Videoconferencing

2.3.1 General

Videoconferencing is an interactive application that allows individuals to communicate with one or several individuals at a time over the Internet using audio, video and data communication and transfer.

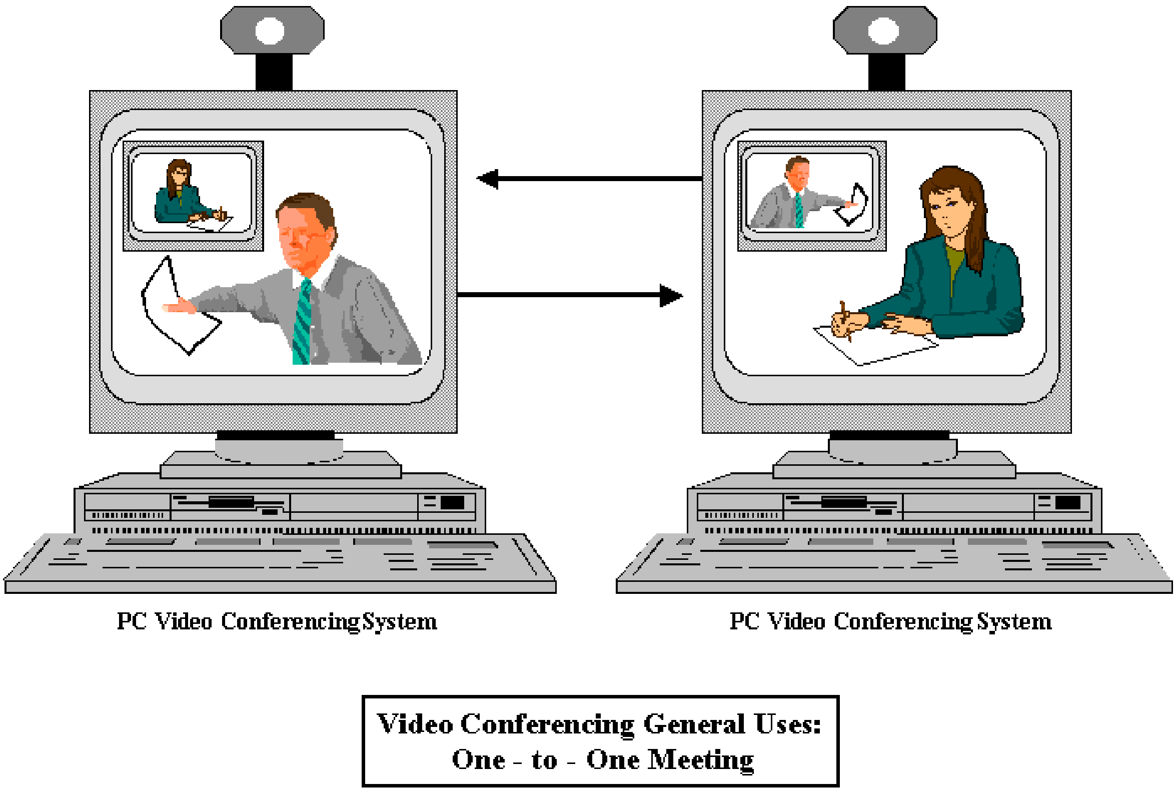


Figure 2-1: Schematic Representation of the Videoconferencing Technology

The basic idea of videoconferencing systems is pretty simple. The end-user looks at a camera, and a video capture device digitises and compresses his image in real-time. He speaks into a microphone, or headset, and a sound card digitises his voice and compresses it into an audio file. He connects to videoconference participants at the other end and transmits his image, voice and other data based on TCP/IP, via modems, LAN-Internet connection, or an ISDN line. The core videoconferencing application lets individuals make and receive calls and manage their voice and image. They may also use other applications to share data with others during the videoconference, such as data files. For instance, one person could launch an application, and the rest of the participants could work on a file in that application via their videoconferencing connection. By the same token, everyone can share a digital whiteboard on which they can write or draw.

A videoconference may be established by either calling the other person directly or connecting over the Internet via an online meeting place. Naturally, the other participant must have his or her videoconferencing application up and running to receive the call. When the call goes through, the applications start exchanging audio and video, and the videoconference begins. A conference can be conducted by using just the audio portion (which requires less bandwidth) or by adding video for a full-fledged videoconference. When video is used, a local video window on-screen is opened where the user is able to see himself and the other participant in a guest video window (Figure 2-1).

2.3.2 Videoconferencing System Requirements

A videoconference system consists of audio-video equipment (monitor, camera, microphone, and speaker) as well as a means of transmitting information between sites. A multimedia workstation (high processor speed and memory) is required. A workstation, 166 MHz Pentium or faster, 32 MB RAM, SVGA graphics card is the minimum requirement, so far (e.g. NetMeeting).

Videoconferencing connections may be limited to a closed network (such as a LAN) or may use public networks (such as regular phone lines). The higher the bandwidth the higher the quality of the course. As an indication, a minimum connection speed of 33.6 kilobits per second (Kbps) is recommended.

2.3.3 Quality Issues and Videoconferencing

There are two groups of problems, which are usually recorded about videoconferencing systems: the video quality and the audio quality. Both are mainly but not exclusively related to the bandwidth of the network and will be explained in the following.

Because of the required resolution and size of the transmitted video, the final video quality might be poor. The higher the video requirements are, the higher its bandwidth required for ensuring the required parameters. When the bandwidth is not high, then the quality of the image drops significantly.

The video frame rate with most videoconferencing systems is mediocre at best. The broadcast standard for TV, for example, is 25 frames per second (fps), which is fast enough to portray smooth, lifelike motion. If the frame rate slows down, motion will look jerky. A tolerable frame rate is around 15fps. At this speed the video contains noticeable pauses, which can be a little distracting. When the frame rate drops further, however, the image grows increasingly jerky and difficult to watch.

Image quality also depends on the resolution of the video window. The two standard resolution sizes are CIF's (Common Interchange Format) 352 pixels by 288 pixels and QCIF's (Quarter CIF) 176 pixels by 144 pixels. Both are much lower than the resolution used in television broadcasting. So even over a fast connection the image will look blocky or blurry. The slower the connection, the worse the image quality will be, in general.

Audio quality is less of an issue because less data is involved. As a rule, the audio quality will be about as good as that from an ordinary telephone line regardless of which transmission scheme it is used. Nevertheless, more bandwidth helps to have sharper and clearer sound.

2.4 Voice over IP

2.4.1 General

Voice over IP (VoIP) is a general term used to incorporate the process of sending voice data in real time over a network, by using the Internet Protocol (IP). The importance of such a process is rather significant for popular applications such as Internet Telephony and Internet Fax Service.

The basic system for transmitting Voice over IP – in a simplified way – is given in Figure 2-2. It includes:

- The gateway on the telephone end: it is responsible for connecting to the regular telephone network (ability to communicate with any phone in the world). A phone line plugs into the gateway on this end.

- The gateway on the computer end: it connects to the Internet world (ability to communicate with any computer in the world). A computer network plugs into the gateway on this end.

Both gateways are able to process the standard telephone signal, digitise it, compress it, form it into packets for the Internet (by using the Internet Protocol) and route it to the destination over the Internet. They can reverse the operation for packets coming in from the network and going out the phone. Both operations (coming from and going out to the phone network) take place at the same time, allowing a full duplex (two-way) conversation.

The gateways perform the following functions:

- Search: When an IP gateway is used to place a call across an IP network, it receives a called party phone number. It converts the call into the IP address of the far end gateway, possibly through a table lookup in the originating gateway or in a centralised directory server.

- Connection: A connection is established between the originating gateway and the destination one, exchanging all set-up, compatibility information and performing any option negotiation and security handshake.

- Digitising: Analogue telephone signals are digitised in a format useful for the gateway.
- Demodulates it in the appropriate format.
- Compresses it for minimising the size of the processed signal.
- Performs decompression and demodulation for dealing with the data packets it receives.

A gatekeeper is the device, which knows how many users are connected to this network and where they are located, as well. It is required to perform the following functions:

- Address Translation: Translation of an alias address to a Transport Address using a table updated via Registration messages.
- Admissions Control: In terms of authorisation, admission requests, rejects and confirm functions.
- Bandwidth Management: Support for bandwidth request, confirm and reject messages.
- Zone Management: All these functions are provided for the devices are registered within the zone (registered endpoints) of the gatekeeper.

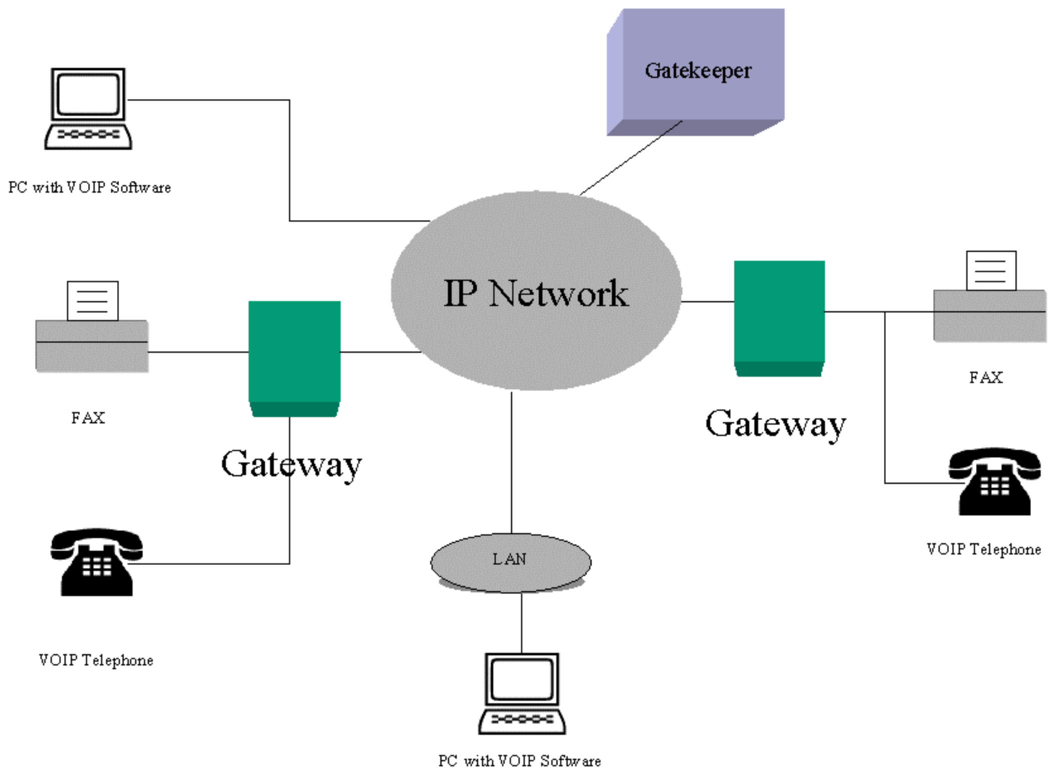


Figure 2-2: Schematic Representation of the Voice over IP Technology

2.4.2 Voice over IP System Requirements

Special IP telephones, PC software phones and PCs with VoIP software may be used for such a service. The important factor for experiencing a good quality of this service is the connection speed of the network (minimum 64 Kbps).

2.4.3 Quality Issues and Voice over IP

In order to use widely Voice over IP it is very important to provide the same or maybe better quality than the existing systems. Therefore it has to be ensured that the transmitted voice has the highest possible bandwidth allocated, in order not to cause hearing problems and to provide a clear conversation transmission. It is also very important to keep any delays implied on the voice transferring mode as low as possible.

A critical aspect for ensuring the Quality of Service in this particular application area is to minimise as much as possible the amount of data to be transmitted by the means of IP. This can be realised by employing specific coding algorithms.

Another critical aspect is related to the time interval required for completing the transmission. This time is possible to be set up to a minimum by configuring the network, in the way that the voice data are given the highest priority for reaching their destination.

The Voice over IP quality is determined by the following factors:

- Codecs. These should give good quality and low delays.
- Latency (delay). Delays are obviously introduced because of the nature of the system. If the delay is higher than 10 ms then the caller hears an echo. The total transmission delays should be kept as low as possible.
- Packet transmission issues: Some data packets may be lost or delayed to arrive to their destination. There should be a good delay management system, enabling packet prioritisation, optimum packet segmentation and so on.
- Bandwidth: The requirements for bandwidth are usually high (at least 64 Kbps). In order to avoid having noticed delays on the side of the users, techniques such as silence suppression (when there is no voice transmitted, less bandwidth is allocated) and insertion of the comfort noise (the use of background noise so when there is no voice the line sounds active), should be used in the most appropriate manner.

2.5 Streaming Media

In simple terms, streaming technology allows a user to view or hear digitised content as it is being downloaded from a stream server. Thus it allows information to be accessed in virtual real time as the data is being transmitted from another location. This comes in direct contrast with the typical procedure of downloading the entire file before accessing it; a process responsible for monopolising tremendous amounts of both time and disk space, generally depending on such factors as the size and nature of the file, modem speed, and Internet connection.

A complete video-streaming system involves all of the basic elements of creating, delivering, and ultimately playing the video content. The main components of a complete video streaming system to accomplish this – encoding station, video server, network infrastructure, and play-back client – are illustrated in the following:

Step 1. Capture: The first step in the process of creating streaming video is to “capture” the video from an analogue source such as a camcorder or VHS tape, digitise it and store it to disk. This is usually accomplished with an add-in analogue video capture card and the appropriate capture software. Newer digital video sources such as digital video camcorders can be captured straight to disk with a “Firewire” capture board without the analogue-to-digital conversion step. The capture card may also support the delivery of “live” video in addition to “stored” video.

Step 2. Edit/Author: Once the video is converted to digital and is stored on disk it can be edited using a variety of non-linear editing tools. At this stage, an authoring tool may also be used to integrate the video with other multimedia into a presentation, entertainment, or training format.

Step 3. Encode: After the video is edited and is integrated with other media it may be encoded to the appropriate streaming file format. This generally involves using the encoding software from the video-streaming vendor and specifying the desired output resolution, frame rate, and data rate for the streaming video file. When multiple data rates need to be supported, multiple files may be produced corresponding to each data rate. As an alternative, newer video streaming technologies create one file that has “dynamic bandwidth adjustment” to the needed client data rate.

Step 4. Serve: The video server manages the delivery of video to clients using the appropriate network transport protocols over the network connection. The video server consists of a hardware platform that has been optimally configured for the delivery of real-time video plus video server software that runs under a specific operating system such as Microsoft Windows NT. Video server software is generally licensed by the “number of streams”. If more streams are requested than the server is licensed for, the software rejects the request.

Step 5. Play: Finally, at the client station the video player receives and buffers the video stream and plays it in the appropriate size window using a VCR-like user interface. The player generally supports such functions as play, pause, stop, rewind, seek, and fast forward. Client players can run stand-alone or can be ActiveX controls or browser plug-ins. They can decode video using software or using hardware add-in decoder boards.

As might be expected, media files are quite large, commanding tremendous amounts of bits and computing power. Aside from the size of the file, maintaining the quality of the service is also problematic. To be used effectively on the Internet, sound quality needs to reach the level of FM radio as well as video quality should be decent and still fit over standard 28.8 Kbps modems. Thus, in order for audio and video files to be sent and played back in “real time”, a variety of compression techniques are used to closely approximate such a signal.

The specific method of audio as well as video compression, is based on the codecs used, i.e. the actual COmpression and DECompression algorithm. Codecs, in general, attempt to compress data through expressing the frequencies and wavelengths in as simple mathematical form as possible and by removing redundant elements. For example, temporal compression simply codes the difference in subsequent samples (for audio) and frames (for video) than the last sample of data. A media file that has been accessed is streamed in small packets of data and is decompressed once it reaches the user.

Current streaming applications use proprietary mechanisms to compress, establish and control the streams, as well as to transport the data. However, while the various industry players all make use of slightly different methodologies, including slight differences in transmitting video, they all essentially follow the same skeletal approach, established by the first launched player ever, namely the RealPlayer, in 1995.

The mechanics of streaming typically begin by a visitor using a Web browser, accessing a Web page and then clicking on a link to a streaming media presentation (1). This presentation, whether a single file or group of files, resides on the host’s server. The host server immediately creates a small metafile and sends it to the visitor’s Web browser (2). Upon downloading, the metafile, notifies the user’s PC which media player (RealPlayer, QuickTime, etc.) is to be used (3). Once the player is opened, it utilises information within the metafile to search for the address of the media presentation mentioned in the link. The player reads the link in the metafile and requests the presentation directly from the host server (4). The host server, upon receiving the request, streams the media files to the player, adjusting for the bandwidth being used (5). The player begins playing the presentation as the initial packets of data are being received, and keeps playing the file as the remaining parts arrive (see Figure 2-3, where the Real Player and Real Server have been used).

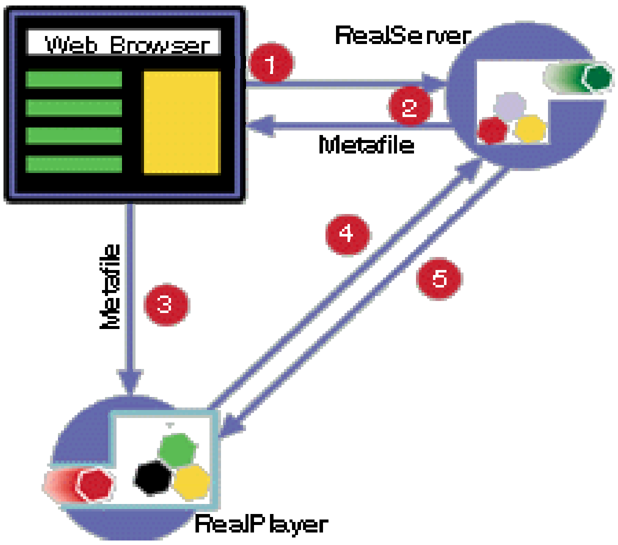


Figure 2-3: Schematic Representation of the Streaming Technology

More specifically, when a user clicks on a link, a bi-directional TCP connection is established between the player and the host server. This connection is used to send control commands, such as when to pause, stop, etc., from the client to the server and vice versa, and is also used for security purposes. Once the request for the presentation is accepted by the server (e.g. RealServer), the requested file is sent via a one-way UDP (User Datagram Protocol) channel. Synchronised Multimedia Integration Language (SMIL) files, are the files that co-ordinate the delivery and playback of the files as they are sent in small packets, essentially acting as a digital conductor. In order to maintain a constant rate of play, audio and video clips are typically buffered, meaning that once UDP packets are read, they are briefly cached, or put in a memory queue, where they are then picked up by the media playback process.

Today, there are numerous players that are shaping the streaming technology (audio and video) landscape, some well-known, and others no doubt to be revealed shortly. The industry leader is largely recognised as Real Networks Inc., which has set the de facto standard in streaming with its RealAudio and RealPlayer, as well as RealVideo products. Apple Computer, Inc.'s QuickTime is also highly utilised as is Liquid Audio and Shockwave. As expected, Microsoft has also had a strong involvement, with an equity stake in Real Networks as well as producing its own technology, Windows Media Player (formerly NetShow). Additional players include such products as Vxtreme, Shoutcast, and Icecast.

At the following, a number of representative cases for exposing the newly introduced video and audio streaming technologies will be presented. Accordingly, there is a case of a movie site (On2.com), a case of an entertainment site (Big Brother), an e-commerce broadband site (Bertelsmann Broadband Group) and the example of an online jukebox (Mediazine) presented.

2.5.1 On2.com

2.5.1.1 On2.com General Information

On2.com is changing the way consumers experience the Internet. A world leader in delivering full-motion, full-screen, television-quality video over the Web, On2 is building a revolutionary entertainment network exclusively for broadband consumers covering such popular interests as movies, music, travel, games and sports.

On2 blends television-quality video with all the interactivity may be expected from the Web, all in full motion, full screen about topics such as:

- Movies (live and archives), massive database of movie and celebrity information;
- Music (movies music, video clips);
- Travel (videos of selected holiday places, stories written by visitors in certain places, online booking);
- Games (trivia about movies, cinema production, actors, celebrities, and so on);

- News (about upcoming movies, details for the productions, events like the Oscar ceremony, festivals, celebrities and so on).

2.5.1.2 On2.com System Requirements

The On2.com minimum system requirements are:

- Broadband connection at a minimum rate of 300 Kbps;
- Operating system of Windows, Mac, or OS;
- Multimedia workstation (a Pentium processor and a minimum of 32 MB memory are recommended);
- Regular Web browser;
- Macromedia Flash / Shockwave;
- Proprietary codec (this is available for free download in the on2.com site).

2.5.2 Big Brother

2.5.2.1 Big Brother General Information

The Big Brother is a totally new interactive television and Internet experience, which has been introduced – or has been planned to be introduced – in Netherlands, Germany, Spain, Italy, Switzerland, UK and USA. The whole idea is based on providing the chance to the user to spy on the everyday and night life of a number of carefully selected individuals, shared a specifically designed house. The residences have to share this house and their life is captured in a number of digital cameras, installed in strategic points of the house when audio is offered as well through a number of microphones. Even in the bedrooms infrared cameras monitor everything that happens at night, thereafter the users witness the residences' life, in full.

The residences have no contact with the outside world; no telephones, television, radio or newspapers are allowed. They have to undertake daily tasks such as baking bread and growing vegetables, in general to complete certain tasks in order to be given some basic luxuries, such as dishwasher and dryers. During the specified time of their stay in the specifically designed house, the users have to right to vote for persons to leave the house on certain periods of time. The aim is only one resident to remain, which will claim a prize. Residents can leave the house anytime, but if they do so they forfeit their right to the prize and must walk away empty-handed.

The users are able to follow the lives of the residences either through a TV channel (a summary of the highlights of the day) or via a live streaming on the Internet, on a twenty-four hours a day basis. They can "control" the lives of the people in the house (thereafter they are given the role of the Big Brother), by voting.

The Internet sites of Big Brother offer:

- Accompanying Web site for the TV show;
- Live video and audio streams;
- Interaction through voting;
- Chat between the users;
- E-commerce (games of Big Brother, T-shirts, and so on);
- Subscription, as selected cameras require "world-line" membership.

2.5.2.2 Big Brother Online System Requirements

The minimum client system requirements for accessing the Big Brother Internet site are:

- Connection of at least 28,8 kilobits/sec;
- Pentium processor and 32 MB memory;
- Multimedia workstation;
- Regular browser.

2.5.3 Bertelsmann Broadband Group

2.5.3.1 Broadband Group General Information

The Bertelsmann Broadband Group offers event broadcasting and video or audio streaming, in terms of:

- Top Film [Action Videos such as series (shows, soap and family, drama, cartoons), comedy (cabaret, satire, slapstick) / Films and Movies (in the categories of drama, romantic, science fiction, adventure, comedy, cartoons, horror and mystery, old and classic) / Erotic (only documentation of soft erotic) / Music (music clips of pop, rock, alternative, rap, hip hop and electronic) / Lifestyle divided into gastronomy (recipes, shop, hot news, guests) and singles and friends (video with persons, contact details, personal information seeking for happiness and love, meeting and friends, leisure time and sports, travelling and adventure)];
- Travel (text and video for specific places, information for travelling by car, train and boats);

- Nature (still videos and documents about people, animals, water, earth, air and scientific theories and news);
- Culture (firms and documentation for literature, religion, music and arts);
- Children matters (films, series, shows and educational activities and games);
- Sport (not available yet in the beta version);
- Highlights of news (not available yet in the beta version);
- News (not available yet in the beta version);
- E-commerce (not available yet in the beta version).

Other services offered are e-mail access and chat-rooms. The system is designed in a way that allows more than one operation to be opened at the same time and offers a full functionality rate similar to the proper video and audio devices.

This service is specifically designed for TCP/IP based cable transmission. It is currently available in a beta version (for Internet access only), which will be tested for a year, before launching the final product in its commercially exploitable form. This final product would be available for access via TV, combined with a set top box with a specifically designed keyboard and a remote control feature.

2.5.3.2 Bertelsmann Broadband Group System Requirements

The BBG Beta version minimum system requirements are:

- Broadband connection at a rate of 1.5 - 2 Megabits/sec (cable modem);
- Operating system of Windows 95;
- Multimedia workstation (a Pentium II processor, 300 MHz and a 64 MB memory, 4 MB graphic card, 17´´ monitor and 2 GB hard disk free space are required);
- Web browser Internet Explorer 5;
- Flash 4 player, Oracle Video Client 3.2.

2.5.4 Music Mediazine

2.5.4.1 Music Mediazine General Information

The Music Mediazine is a prototype application for interactive broadband technologies. It provides audio and video processing facilities, as well as:

- Music videos by Bertelsmann Music Group (BMG) and special BmS/ productions;
- Audio music from BMG;
- Text format information concerning various artists;
- Lyrics of popular songs and poems;
- Video and text chat-rooms;
- Voting on popular songs (thereafter the top list is produced);
- E-commerce.

2.5.4.2 Music Mediazine System Requirements

The recommended requirements for the Mediazine user are:

- Pentium II 450 MHz PC, 128 MB RAM, 16 MB video RAM;
- Suitable video capture board and camera;
- Full-duplex sound card;
- Internet Explorer 5;
- Proprietary codec (Real Player or Windows Media Player or QuickTime);
- Internet access well above 128 Kbps, at least 300 Kbps available bandwidth is recommended, for good quality performance.

2.5.5 Quality Problems on Streaming Media Reported by the End-user and Explanation Given by the Provider

The problems recorded by the end-users can be grouped into the following categories:

a. General problems

These may be problems related to the troubleshooting on the installation process of the plug-ins. These can be solved usually by the instructions given by the provider of the special software.

Another frequently recorded problem is that the pages are loaded in a long time. This is due to inadequate connection speed, or inadequate allocation of resources for serving the unpredicted large number of page viewers. The provider can offer the solution to this problem by allocating more resources for serving the increased number of the page viewers.

b. Related to the video

Users complain that the live stream sometimes does not come at all or that the live stream picture flicks. This is a problem related to the insufficient data rate and indicates the need for optimisation of the used graphic cards.

c. Related to the sound

Users argue that the sound of the streams is not sometimes good or the sound sometimes does not agree with the picture or whenever different sound channels are selected the sound is the same everywhere. Since providers argue that they face numerous technical problems on the sound distribution on the selected spots thereafter they are able to solve this problem by optimising their used distribution techniques.

2.5.6 General QoS Related Problems on Streaming Media

When dealing with streaming audio and video, the sound quality is found to be a direct function of modem speed, Internet connection, the playback mechanism and speakers used, and the codec algorithms utilised by the software. In general, the slower the speed and connection, the lower the quality is. While most codecs do try to adjust for different connection speeds, success varies. In addition, the complexity of the coding and decoding algorithm dictates how much computing power is needed and the speed of the operation. This can be critical due to the natural time constraints for conversion during real-time playback, despite buffering. Additionally, depending on how the audio data is coded and streamed, lost information in transfer may not always be retransmitted in time, thereby causing gaps. Finally, perceived audio quality of a highly compressed signal is a function of both the range of sound frequencies reproduced and the accuracy in representation of the waveform of the original. As bandwidth capabilities are largely outside of their control, these are the main issues that streaming technology companies are currently focused on.

The quality of digital video may be judged based on three main factors:

1. Frame Rate. The number of still pictures displayed per second to give the viewer perception of motion. This should be set up properly to ensure that there is a desirable and constant rate of still pictures displayed per time interval;

2. Colour Depth. The number of bits per pixel for representing colour information. For example, 24 bits can represent around 16.7 million colours, 16 bits 65536 colours, or 8 bits only 256 colours.

3. Frame Resolution. Typically expressed as the width and height in pixels. For example, a full screen PC display is 640x480; a quarter screen is 320x240, a one-eighth or “thumbnail” is160x120. Generally speaking the bigger the display the lower the quality of the picture.

Consequently, the general QoS related problems for video and audio streaming are defined as:

- Small resolution;
- Jitters during the transmission of data;
- Objects do not move smoothly;
- Audio drop outs;
- Relatively small screen projection;
- When the full screen output is selected the video quality drops significantly;
- Vertical and horizontal triggers;
- Limited bandwidth.

2.6 Significant Factors for Multimedia Internet Services

The most critical factor for offering Multimedia Internet Services is the quality. Quality may be determined mainly on the side of the end-user, although the service provider is the main responsible for that. The most significant parameters for determining the quality on the side of the end-users, in terms of offering Multimedia Internet Services are:

- The Connection speed (bandwidth of network), which is responsible for the speed of exchanging information between the user and the Internet. The possible types of connection speed are shown in Table 1. Various European counties support different connection modes and provide their own values within the typical range described in the same table.

- The traffic between the user and the source ends, which is important for ensuring the standard connection speed. A heavily condensed network results on the realisation of a low connection speed.

- The day and local time of connection in both user and source ends, which is able to slow down the time for accessing the streams, as in different days and times of the day there is a difference in terms of traffic in the computers networks.

- The resources of the client workstation, which can become rather critical for the presentation of the streams and the connection itself. A powerful multimedia workstation with a strong processor and lots of memory can be more satisfactory for the presentation of streamed files.

- The perception of the overall quality of the services, which is mainly subjective. It depends on:

  ⇒ age (in principle young users demand more),

  ⇒ gender (male are more oriented on technology),

⇒ location (in terms of the infrastructure and the connection lines supported),

⇒ the Internet experience (an experienced user is harder to satisfy than a non-experienced one) and

⇒ skill level (a high-skilled user demands more quality) of the individual user.

Connection Type	**Speed**	**Used by**
Mobile Phone	9.6 to 14.4 Kbps	All Users
Modem	28.8 to 56 Kbps	Home Users
ISDN	64 to 128 Kbps	Business and Home Users
DSL	128 Kbps to 2 Mbps (upstream), up to 8 Mbps (downstream)	Home and Business Users
Cable Modem	Up to 2 Mbps upstream	Home Users
T1	1 to 2 Mbps	Business Users
T3	30 to 50 Mbps	ISPs, Universities, Business Users

Table 2-1: General Description of Available Connection Modes

The end-user is able to control the most of the above parameters, in order to improve the perspective overall quality. He is unable though to influence the standard connection speed, which is provided at any time and it is closely connected to the problem of the limited bandwidth. In order to overcome this problem, a substantial amount of video data compression is necessary to be used by the service providers. Successfully delivering digital video over limited bandwidth networks can involve processing the video using three basic methods:

- Scaling the video to smaller window sizes. This is especially important for low bandwidth access networks such as the Internet, where many clients have modem access. Techniques to lower this further involve scaling one or more of the three factors mentioned in the previous sections: frame rate, colour depth, and frame resolution. For example, scaling the frame resolution results in different size windows for showing the video on the screen. Further scaling of all three parameters can dramatically reduce the video rate.

- Compressing the video using loose compression techniques. This is generally needed for almost all networks because of the high bandwidth requirements of uncompressed video. Different algorithms and techniques – “codecs” – have been developed for compressing video signals. Video compression techniques take advantage of the fact that most information remains the same from frame to frame. Taking advantage of this, enables the video information to be represented by a “key frame” with “delta” frames containing the changes between the frames. In addition, individual frames may be compressed using loose algorithms similar to JPEG photo-image compression.

- Streaming the video using data packets over the network. Small video files may be downloaded and played, but there is a tendency to stream larger video content for faster viewing.

2.7 New Products

2.7.1 RealProducer 8 Plus

RealProducer 8 is the latest version of the encoding tool produced by RealNetworks, designed to create audio and video streaming files. Whether the source is a digital file or a live feed directly to the server, the encoder can convert it into a RealMedia format. Before the file is encoded the user is able to define some parameters in regard to the quality the end-user will to receive. In this latest version the compression rate has been improved. The quality is much better than for instance in MPEG-1 or MPEG-2.

This new product includes the powerful tool of SureStream. With the new SureStream recording the widest possible audience can be reached, as it is guaranteed that all end-users would be able to enjoy the service, with the best listening and viewing quality, specifically optimised for their bandwidth.

There is a number of advantages to using SureStream, such as:

A single RealMedia clip can be recorded for multiple target audiences (up to seven versions);

A clip can be automatically switched to a lower bandwidth, when poor network conditions are valid.

SureStream RealMedia files can combine several different streams that take advantage of a number or even all of these features. For example, you can record a video clip for both 28 Kbps and 56 Kbps audiences, and RealPlayer will automatically use the correct stream based on the end-user’s connection speed.

Furthermore, RealProducer 8 Plus has the bandwidth-simulator integrated. It is possible to simulate the end-users’s connection speed for testing the encoded RealStream.

2.7.2 On2.com True Motion VP 3.2

The True Motion VP 3.2 technology promises to provide full screen TV quality streaming video and audio over the Internet at data rates as low as 200 kilobits per second.

VP 3.2 combines state of the art advances in video compression, server-side streaming and bandwidth management to compensate for bottlenecks in the Internet in infrastructure, responsible for reducing the quality when an increasing number of broadband users come online.

3 Requirements of Legacy Applications

In this chapter, the applications selected for the first trial are analysed. The application requirements towards the network are precisely analysed as well as QoS tests/validation scenarios at application level (respectively at end-user level) and at network level. Note that the difference between the two kinds of tests/validation is quite important and can signify totally different tests/validation scenarios.

Extract from [Adanez]: *"Being network-oriented or user-oriented the concept of quality of service is very different. (...) Performance is shifting from a network-centric perspective to an application-centric or a user-oriented perspective. This shift has not been followed by measurement tools.*

Quality of Service Criteria – the metrics –

One of the main problems of evaluating QoS is to define what QoS means. This acronym has been used for several years without a clear definition of its meaning hence leading to dozens of interpretations of it. Clearly, QoS does not support a single definition because it depends on who the QoS is directed to and on who decides of the level of quality he is receiving. Quality is neither a quantitative metric nor an objective one.

Therefore, to measure QoS it is necessary to previously define exactly what are the criteria in terms of metrics and quantities. Today SLA contracts are based on traditional network metrics namely; delay, loss, throughput and availability. The names of the metrics may change slightly such as response time or latency instead of delay but the criteria remains the same. These four parameters deliver a quantitative network view of QoS that an IT manager can use. However, showing that the network delivers a given throughput does not make any sense to a corporate user. Users perceive QoS based on a different set of qualitative criteria and metrics. What an IT manager generally hears as a QoS value is 'the network is slow!'. How can this be quantified ? How is it possible to map loss, delay and throughput to such a user-oriented criteria ? The solution is not simple if solution at all."

This chapter should help us to get a better understanding of the chosen applications and determine

- what their requirements towards the network are and which parameter should be considered by testing, and
- what the user-oriented criteria could be, or how the quality could be validated at application level.

3.1 Streaming Media: RealSystem

3.1.1 Description

The Real System is a three-part client/server system for streaming media. It mainly includes three modules produced by RealNetworks (http://www.realnetworks.com):

- **RealServer:** The RealServer serves all the RealMedia files, which are prepared by the RealProducer. It is possible to stream live video/audio or some encoded files. The newest version includes the powerful tool of SureStream. With the new SureStream streaming the widest possible audience can be reached, as it is guaranteed that all end-users would be able to enjoy the service, with the best listening and viewing quality, specifically optimised for their bandwidth. There is a number of advantages in using SureStream, such as:

    - A single RealMedia clip can be recorded for multiple target audiences (up to seven versions).

    - A clip can be automatically switched to a lower bandwidth, when poor network conditions are valid.

  SureStream RealMedia streams can combine several different streams that take advantage of any or all of these features. For example, you can record a video clip for both 28 Kbps and 56 Kbps audiences, and RealPlayer will automatically use the correct stream based on the end-user's connection speed.

- **RealProducer:** RealProducer 8 is the latest version of the encoding tool, designed to create audio and video streaming files. Whether the source is a digital file or a live feed directly to the server, the encoder can convert it into a RealMedia file. Before the file will be encoded, the user is able to define some parameters in regard to the quality the end-user wanted to receive. Also the SureStream technology is included, so it is possible to create streams.

- **RealPlayer:** The RealPlayer can play all types of RealMedia files and the most of other used file types. It works either as a plug-in or as a standalone player and is launched whenever a media file is encountered. The RealPlayer 8.0 is necessary to play stream files which were created with SureStream.

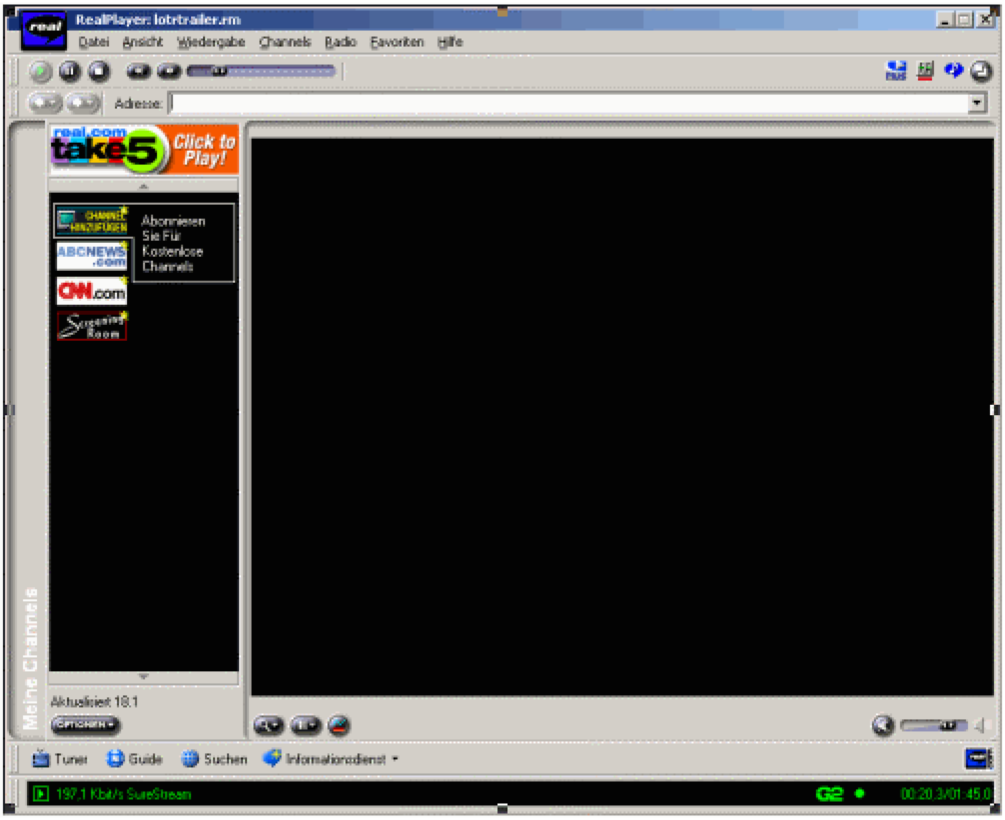


Figure 3-1: Screenshot of RealPlayer

Typical usage scenarios are the classification of the different network connections. There are **five** important network speeds at the moment: modem, ISDN, dualband ISDN, cable/DSL, or LAN (10 Mbps) connection. The picture size depends on the size of the original video stream, and will not changed during the encoding.

3.1.2 Configuration

The RealPlayer need the information, which network speed is provided. This will be set during the installation or after that in the regular preferences (see Figure 3-2).

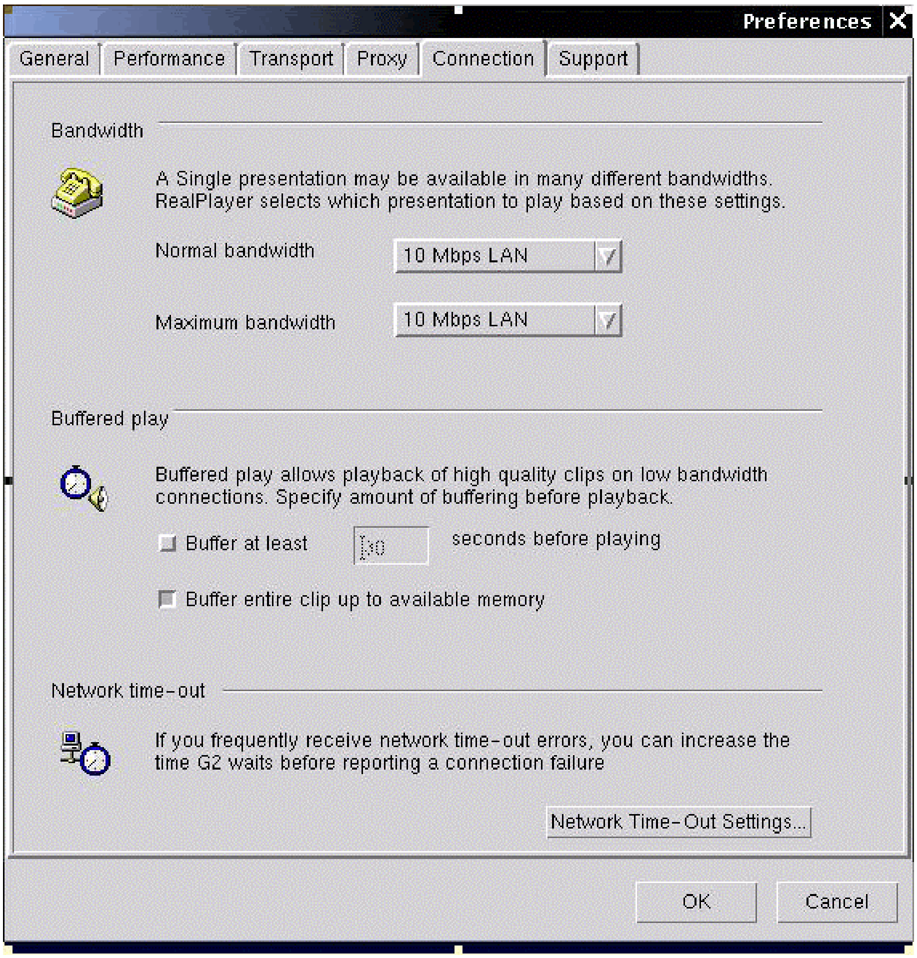


Figure 3-2: Screenshot of RealPlayer preferences

3.1.3 Technical requirements

Application: RealSystem					
Version:					
Comments: The RealSystem includes the RealServer and the RealPlayer. Free RealServer and RealPlayer LINUX / Windows: http://www.realnetworks.com					
Services components					
Video with Audio					
Network characteristics					
TCP - Port: 554					
UDP - Port : 6970 (adjustable)					
Technical characteristics					
Supported RTSP RealTime Streaming Protocol					
Default codec (Real Video 8): RV300					
QoS specification					
(max) Delay in ms	A higher delay is possible, because the RealPlayer is adjustable to buffer some data for reducing the problems with delay.		**NOTE:** it depends on the selected buffer size and the minimum bandwidth		
(max) Jitter in ms	The RealPlayer reduces automatically the bit stream, if the bandwidth is to low for the currently used data transfer rate.		**NOTE:** it depends on the selected buffer size and the minimum bandwidth		
(max) Loss	< 10% the quality is acceptable (UDP)	> 10% the video stops temporarily	**NOTE:** it depends on the selected buffer size and the minimum bandwidth		
Ordering of packets	Important				
BW guarantee	Depends on size of the video				
Traffic description					
Session Characteristics	Modem	Single ISDN	Dual ISDN	Cable/DSL	LAN
Bandwidth	28.8 Kbps	56 Kbps	128 Kbps	256 Kbps	1 Mbps
Packet size average (Byte)	360	421	488	751	755
	Packet Size goes from 11 up to 1427 Byte				
BSS in bytes					
(m)					
EAR	Depends on selected Network speed and video				
PR in bit*s^{-1}					
SR in bit*s^{-1}					
Possible traffic model					
Reservation style					
	p2p	p2p	p2p	p2p	p2p
Possible network services					
	PMM	PMM	PMM	PMM	PMM

Table 3-1: RealSystem requirements

3.2 Multimedia Conferencing: NetMeeting

3.2.1 Introduction

NetMeeting (NM) (Figure 3-3) is a Windows-based conferencing tool that provides real-time communication over the Internet. NetMeeting supports audio, video, and data capabilities, thus allowing users to talk and see each other, work together on virtually any Windows-based program, write on an electronic whiteboard, transfer files, or use the text-based Chat program.

The usage of NetMeeting is fairly simple. The user can launch the application and wait for incoming calls, or call the person with whom he would like to communicate. Calling a person (*Call→New Call*) can be achieved in two ways: either by entering the IP address of the callee, or by searching the name in the Internet Directory.

After initiating the conference, audio is transmitted by default. The user can disable audio through the use of the Resource Kit Wizard. Moreover, NetMeeting can be set up to automatically send/receive video. Data collaboration can be initiated through the Tools menu.

Figure 3-3: Screenshot of NetMeeting

3.2.2 Configuration

There are some options regarding network speed, audio and video, that affect the consumed bandwidth. These can be configured both during and after the application's installation.

To start with, the network speed, is a very significant parameter for NetMeeting as it can affect the streaming performance of the application, by determining the codecs to be used for audio and video. The network speed is set through the *Tools→Options→General-Tab→Bandwidth Settings...* menu. There are four different predefined values for the total bandwidth, which correspond to the underlying physical medium. The user selects the used medium and NetMeeting determines the total (for audio, video, and data) bandwidth automatically (e.g. 435.19 Kbps over LAN).

- 14.4 kilobits per second (Kbps) modem
- 28.8 Kbps or faster modem

- Cable, xDSL, or Integrated Services Digital Network (ISDN)
- Local area network (LAN)

Audio is configured through the *Tools→Options→Audio Tab*. There are the following options in the Audio Tab:

- *Enable full-duplex audio so I can speak while receiving audio*. Audio can be sent and received simultaneously, provided that the sound card supports full-duplex audio. An identified problem with Windows NT is that although many sound cards provide full-duplex audio, only a few sound cards drivers have this same capability.
- *Enable auto-gain control*. Auto-gain is a feature of the sound card and driver that increases and decreases microphone volume, depending on how softly or loudly the user is speaking.
- *Automatically adjust microphone volume while in a call*. This is a feature of the application that performs a similar operation to auto-gain. It is included because not all sound cards support auto-gain.
- *Enable DirectSound® for improved audio performance*. DirectSound provides the ability to play multiple audio streams simultaneously.
- *Tuning Wizard*. It directs the user to select the sound card and connection type, and to tune audio volume.
- *Advanced*. Through this GUI, the user can adjust the audio compression settings. Although NetMeeting determines the default setting (G.723.1), the user can manually select one of the other codecs.
- *Silence Detection*. Silence detection pauses audio transmission if the microphone does not detect any sound. If *Adjust silence detection automatically* is chosen, NetMeeting performs this operation automatically (default). If *Let me adjust silence detection myself* is selected, the user can set the level of the sound to pause audio transmission.

The *Tools→Options→Video Tab* controls the transmission of video. The user can set the following options:

- *Sending and receiving video*. The user can select to send and receive video automatically at the beginning of each call.
- *Send Image Size*. There are three selections on the size of the transmitted video image: small, medium, and large. This parameter heavily affects the required bandwidth.
- *Video Quality*. The user can set the quality of the video by moving the tab between *Faster video* and *Better quality*. This parameter also heavily affects the consumed bandwidth.

- *Video Camera Properties*. The user can adjust brightness, hue, and saturation through the source and format dialog boxes provided by the manufacturer of the device (if supported by the device's driver).

3.2.3 NetMeeting bandwidth considerations

The audio, video, and data subsystems produce independent flows for network transmission at their own rates. The audio stream is fairly constant when speech is sent. The video and data streams are not constant, as they depend on the amount of the sending content, as well as to other parameters. However, NM is able to control the total amount of bandwidth sent, through a mechanism called *Intelligent Bandwidth Management and Control*. It always prioritises audio over data over video.

NetMeeting can control the bandwidth consumption for audio and video by defining an average value for the combined streams. This average value can be set up (anywhere between 85 to 621 Kbps) through the NetMeeting Resource Kit Wizard. The Resource Kit Wizard is not contained in NetMeeting, but it has to be downloaded and installed separately.

According to Microsoft, the following factors affect the amount of network bandwidth that audio and video conferencing scenarios use:

- The amount of motion for video; more motion equals higher bandwidth.
- The size of the video window; small, medium, or large.
- The quality of the video; faster video or better quality.
- The type of camera used (black-and-white or colour); the more CPU the camera uses, the slower the performance, and the less bandwidth used.
- The microprocessor type and speed, and the network speed.
- The OS version, the video adapter, the available RAM, etc.

Table 3-2 presents the average and maximum bandwidth usage for NetMeeting under various operating scenarios. The tests have been performed under both Windows 98 and Windows NT 4.0 operating systems. We can observe that sometimes there are significant variations between the two versions of the OS.

Scenario description	Average bandwidth (Kbps)	Bandwidth range (Kbps)
	Win98 / Win NT 4.0	
Half-duplex audio conversation	19 / 24	0-62 / 0-81
Full-duplex audio conversation	38 / 38	0-90 / 0-66
Small window and low-quality video	49 / 75	0-162 / 0-152
Medium window and medium-quality video	130 / 130	0-230 / 0-245
Large window and high-quality video	700 / 790	0-900 / 0-900
Medium window and high-quality video - normal movement	540 / 600	0-710 / 0-820
Medium window and high-quality video - increased movement	625 / 775	0-775 / 0-880
Full-duplex audio and medium window/high-quality video	550 / 600	0-795 / 0-840

Table 3-2: NetMeeting Bandwidth Tests

Results regarding audio:

- Using the G.723.1 audio codec, NetMeeting consistently generates predictable bandwidth results.
- The bandwidth deviations are due to several variables, including conversion pauses and standard network operation, which queues up audio data and sends it as packets. The maximum value is always observed at the beginning of the session, and it has a very short duration.

Results regarding video:

- The graph of bandwidth consumption for video contains some **spikes** that occur every **15 seconds** when NetMeeting sends a complete video frame. If transmission occurred over a modem, video could be scaled back to accommodate the available bandwidth (slower refresh rate than once every 15 seconds).

3.2.4 Technical requirements

NetMeeting makes use of the H.323 protocol. H.323 provides the core technologies for multimedia teleconferencing over non-guaranteed bandwidth packet switched networks (like Ethernet).

NetMeeting uses different IP sessions for audio, video, and data. The operations performed during call set up, for the creation of the actual audio and video streams are depicted in Figure 3-4. The only a priori known port is the standard 1720 port, which the H.323 protocol uses to initially connect to the callee. All other ports are ephemeral, i.e. they are dynamically negotiated through the Q.931 and H.245 protocols. The resulting audio and video ports in both parties (caller and callee) can only be identified by introducing an **H.323 proxy**. This also implies that the NetMeeting application must be launched and a call must be initiated before we can be aware of the used ports.

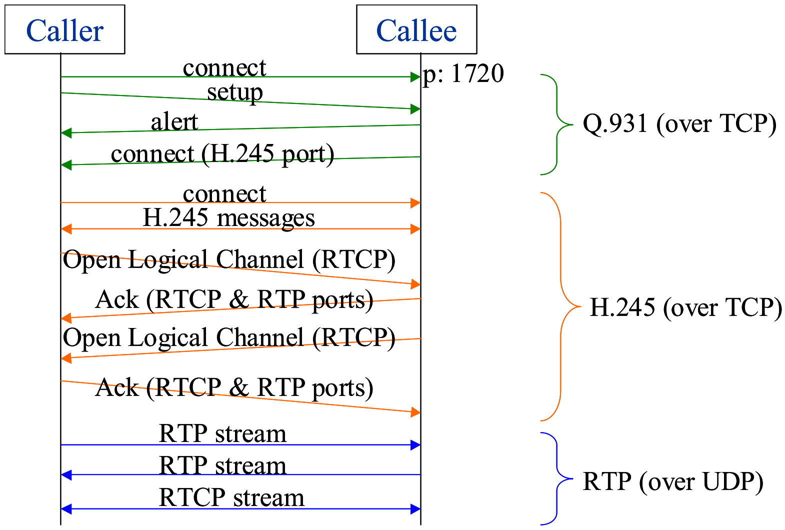


Figure 3-4: NetMeeting call set up procedure

3.2.5 Session characteristics

We can identify three service components in NetMeeting: **Audio**, **Video**, and **Data**.

For the first trial, NetMeeting will provide audio and video streams to test the corresponding network services. Because of the intelligent bandwidth management features of NetMeeting and the combination of streams' bandwidth, we would propose to disable (i.e. not to use) the data subsystem at all.

We can also assume that during the videoconference, the video motion will be probably low. So, the factors that will affect our selection of QoS parameters are: the network connection, the size of the window and the quality of the video. However, as the bandwidth consumption depends also on the specific running environment, a good approach towards the trials is first to run the various NM scenarios *without* any QoS support in order to *measure* the required bandwidth. Sequentially, we can use the measured values (probably compared and combined to ours), to request these specific values through the Reservation GUI.

The template found in Table 3-3 presents general parameters of NetMeeting, as well as specific QoS parameters complying with the combinations of the aforementioned factors. We cannot measure the effect of each factor to the bandwidth independently of the other factors. Therefore, only the combined sessions are reported here.

The template covers the following cases:

- Audio (only), half- and full- duplex
- Video (only), 3 setting: Low quality (small window, low-quality), Medium quality (medium window, medium quality), Good Quality (large window, best quality)

They cover the case that the network setting is LAN (or xDSL).

The selected Network Service is **PCBR** for audio and **PVBR** for video. We could also try audio over the PVBR service. For PCBR, the parameters that have to be configured are: **PR** (0 - 200 Kbps) and **m** (40 - 256 B). There are default values for **M** (= 256 B) and **BSP** (= 256 B). An optional parameter is **EAR**. For PVBR, the parameters that have to be configured are: **PR** (0 - 1 Mbps), **SR** (0 - PR), **BSS** (M-30M), and **m** (40 B - M). There are default values for **M** (= 512 B) and **BSP** (= 1024 B). An optional parameter is **EAR**.

Application: NetMeeting					
Version: 3.01					
Comments:					
Services components					
Audio					
Video					
Data – not taken into account					
Network characteristics					
Port 1720 is used for H.323 setup. All other ports are ephemeral. A Proxy is needed.					
Audio and video streams are RTP streams (UDP)					
Technical characteristics					
H.323 conference protocol					
Supported Video codecs: H.261, H.263 (default)					
Supported Audio codecs: G.711, G.722, G.728, G.723 (default), G.729					
QoS specification					
(max) Delay in ms					
(max) Jitter in ms					
(max) Loss					
Ordering of packets					
BW guarantee					
Traffic description					
Session Scenario	Audio half-duplex	Audio full-duplex	Video Low Quality	Video Medium Quality	Video Good Quality
Bandwidth					
Packet size					
BSS in bytes	-		? (10xM = 5120)	? (12xM = 6144)	? (16xM = 8192)
(m)	80 (60)	80 (60)	80	80	80
EAR in bps	20 K	40 K	75 K	130 K	700 K
PR in bps	60 K	90 K	160 K	240 K	900 K
SR in bps			? (75 K)	? (130 K)	? (700 K)
Possible traffic model					
	TCL1	TCL1	TCL2	TCL2	TCL2
Reservation style					
	p2p	p2p	p2p	p2p	p2p
Possible network services					
	PCBR	PCBR	PVBR	PVBR	PVBR

Table 3-3: NetMeeting requirements

To be noted here, that these numbers are only indicative, and could lead to false results during the trials. The most appropriate procedure is to measure the bandwidth (and possibly other) characteristics of NetMeeting in the trial reference environment, without QoS support. From these measurements, the most appropriate values for the AQUILA traffic classes must be derived, and requested through the EAT.

3.3 IP Telephony: WinSip

3.3.1 Description

3.3.1.1 WinSip

WinSip is developed by Siemens AG Österreich and the Institut für Computertechnik Technische Universität Wien. This is an IP Telephony software component based on the Session Initiation Protocol (SIP). SIP is an IETF recommendation [RFC2976].

With WinSip it is possible to create, modify, and terminate audio sessions with one or more participants. WinSip can invite parties to join both unicast as well as multicast audio sessions, and it also has the added functionality that the initiator does not necessarily have to be a member of the audio session to which it is inviting.

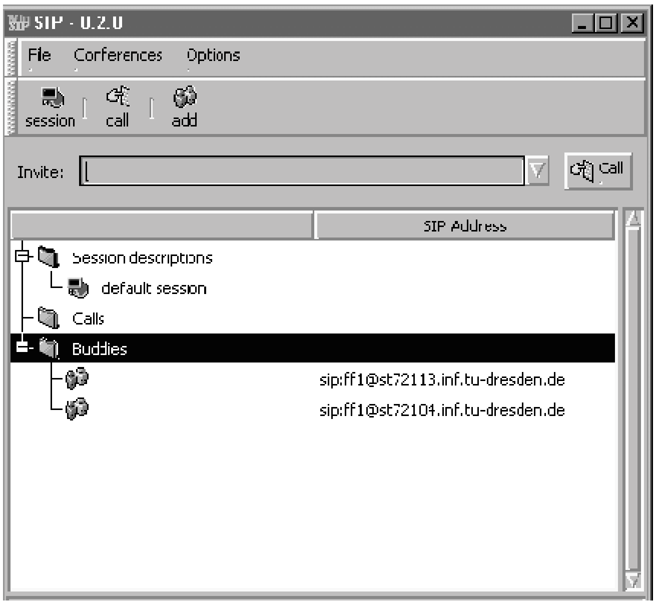
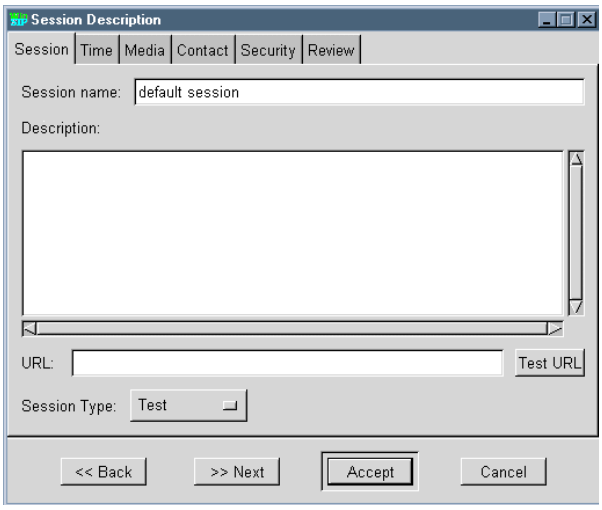


Figure 3-5: Screenshot of WinSip

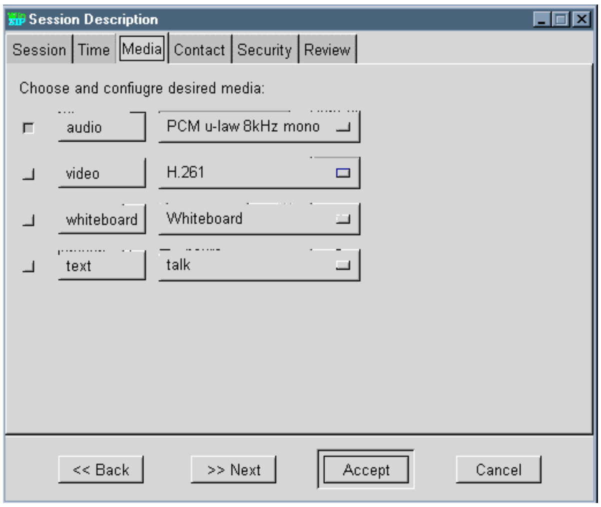
With WinSip it is planed to manage

- Sessions:

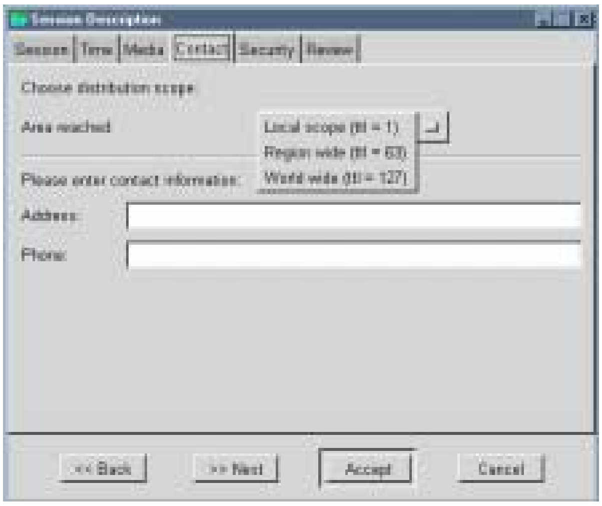


a. Session information

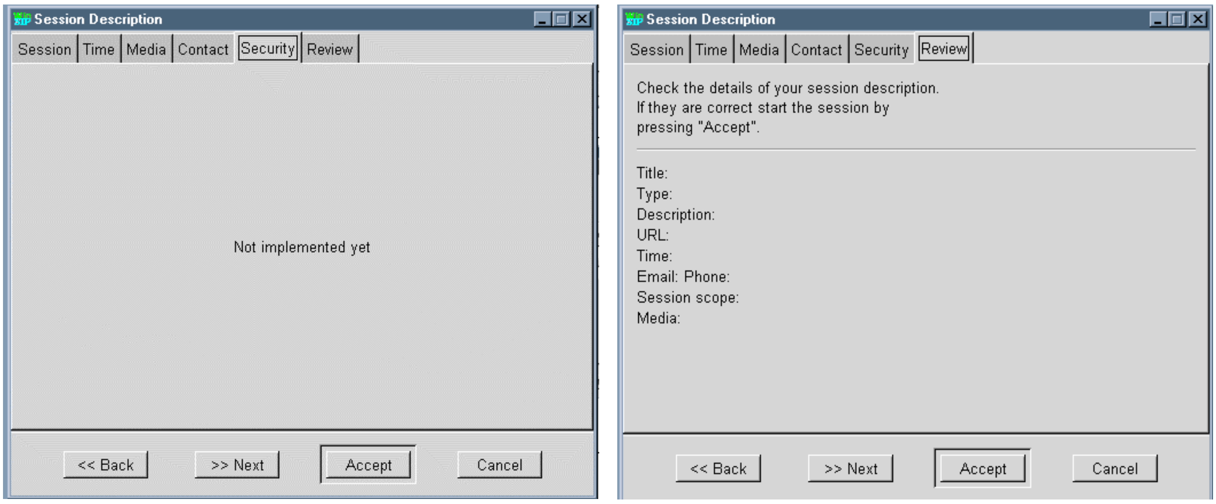
b. Time information



c. Media information



d. Contact information



e. Security information

f. Review information

Figure 3-6: Screenshots of the different components of a session description

- calls, and
- buddies: information on the contact person

3.3.1.2 Significance of VoIP applications

Voice is the main information medium, and voice traffic was the predominant use of the telecommunication networks for most of the last century. Voice traffic continues to grow even though it represents a decreasing percentage of overall telecommunications traffic compared to data [ITU2001]. Support for IP-related technologies is now a strategic element in the design, development and use of telecommunication networks.

In comparison with the telephone networks (engineered to provide real-time or synchronous, two-way voice conversations), at the origin the Internet Protocol has been designed and developed for not real-time and asynchronous communications referred to as connectionless or stateless. There is no total control of traffic management, the default packet delivery is the “best-effort” one, and end-to-end quality cannot be guaranteed. For these reasons, the traditional Internet is not well suited to carry a voice telephony service, which cannot tolerate more than minimal transmission delays (precisely presented in 3.3.1.4). Nevertheless an action performed in Hungary, where the consumers have had the choice of using VoIP since 1999, shows the attraction for VoIP. In fact VoIP is 20-50% cheaper than a conventional circuit-switched call [ITU2001]. The choice to use VoIP is an economically rational one even if quality problems were reported, and VoIP does not always match user expectations. At the present consumers must make a trade-off between price and quality.

The ITU report [ITU2001] reports as well results of the TeleGeography Inc. estimating that in 2000 3% of the global total of international traffic were carried over IP-based networks and that the market is growing fast. The IP telephony market place is of high significance for the future even if VoIP at the moment, because of the lack of quality, is not well accepted by the consumer.

Therefore AQUILA aims to support application services that require "real-time" transport such as audio and video streams. VoIP is one example of interactive, real-time audio between users.

3.3.1.3 Applications implementing IP Telephony

Extract from [VoIP99]:

a) *"Public Switched Telephone Network (PSTN) gateways: Interconnection of the Internet to the PSTN can be accomplished using a gateway. A PC-based telephone for example would have access to the public network by calling a gateway point close to the destination.*

b) *Internet-aware telephones: Ordinary telephones can be enhanced to serve as an Internet access device as well as providing normal telephony.*

c) *Inter-office trunking over the corporate intranet: replacement of tie trunks between company owned PBX's using an Intranet link would provide for economies of scale and help to consolidate network facilities.*

d) *Remote access from a branch office: A small office could gain access to corporate voice, data, and facsimile services using the companies Intranet services.*

e) *Voice calls from a mobile PC via the Internet: Calls to an office can be achieved using a multimedia PC that is connected via the Internet. One example would be using the Internet to call from a hotel instead of using expensive hotel telephones.*

f) *Internet call center access: Access to call center facilities via the Internet is emerging as a valuable adjunct to electronic commerce applications. Internet call center access"*

3.3.1.4 VoIP applications and quality

Extract from [Watson]:

"Audio and video information is sent over the Internet in a digital fashion in small blocks known as 'packets'. These packets can be delayed or lost because of congestion on the MBone.

Congestion occurs at the network routers through which the information must be passed in order to be sent to the correct destination. If information is held up for too long at the routers, the packets may arrive too late to be played out at the receiving end. Alternatively, if congestion at the router is very great, some packets may get dropped at that point. The end result of these two occurrences is the same: packet loss.

Packet Loss

Packet loss is the single-most disruptive factor in multimedia conferencing over the Internet. This is especially true with respect to the perception of speech. Video quality is also impaired by packet loss, but its effects are less harmful to the communicative process, since it is audio that is widely considered to be the most critical component […]. When packets of speech information are lost, the effect on listener perception is dependent on the size of the packet and the loss rate […]. Audio packets sizes are of 20 ms, 40 ms, or 80 ms duration. The smallest meaningful element of speech, the phoneme, has an average size of 80-100 ms […], and so losses of this size can interfere with the intelligibility of the perceived speech."

Extract from [VoIP]:

"The quality of sound reproduction over a telephone network is fundamentally subjective, although standardised measures have been developed by the ITU. It has been found that there are three factors that can profoundly impact the quality of the service:

***Delay**: Two problems that result from high end-to-end delay in a voice network are **echo** and **talker overlap**. Echo becomes a problem when the round-trip delay is more than 50 milliseconds. Since echo is perceived as a significant quality problem, VoIP systems must address the need for echo control and implement some means of echo cancellation. Talker overlap (the problem of one caller stepping on the other talker's speech) becomes significant if the one-way delay becomes greater than 250 milliseconds. The end-to-end delay budget is therefore the major constraint and driving requirement for reducing delay through a packet network.*

***Jitter (Delay Variability)**: Jitter is the variation in inter-packet arrival time as introduced by the variable transmission delay over the network. Removing jitter requires collecting packets and holding them long enough to allow the slowest packets to arrive in time to be played in the correct sequence, which causes additional delay. The jitter buffers add delay, which is used to remove the packet delay variation that each packet is subjected to as it transits the packet network.*

***Packet Loss**: IP networks cannot provide a guarantee that packets will be delivered at all, much less in order. Packets will be dropped under peak loads and during periods of congestion (caused, for example, by link failures or inadequate capacity). Due to the time sensitivity of voice transmissions, however, the normal TCP-based re-transmission schemes are not suitable. Approaches used to compensate for packet loss include interpolation of speech by re-playing the last packet, and sending of redundant information. **Packet losses greater than 10% are generally not tolerable**.*

Maintenance of acceptable voice quality levels despite inevitable variations in network performance (such as congestion or link failures) is achieved using such techniques as compression, silence suppression, and QoS-enabled transport networks. Several developments in the 1990s, most notably advances in digital signal processor technology, high-powered network switches, and QoS-based protocols, have combined to enable and encourage the implementation of voice over data networks. Low-cost, high-performance DSPs can process the compression and echo cancellation algorithms efficiently.

Software pre-processing of voice conversations can also be used to further optimise voice quality. One technique, called silence suppression, detects whenever there is a gap in the speech and suppresses the transfer of things like pauses, breaths, and other periods of silence. This can amount to 50-60% of the time of a call, resulting in considerable bandwidth conservation. Since the lack of packets is interpreted as complete silence at the output, another function is needed at the receiving end to add 'comfort noise' to the output.

Another software function that improves speech quality is echo cancellation. As was noted earlier, echo becomes a problem whenever the end-to-end delay for a call is greater than 50 milliseconds. Sources of delay in a packet voice call include the collection of voice samples (called accumulation delay), encoding/decoding and packetizing time, jitter buffer delays, and network transit delay. The ITU recommendation G.168 defines the performance requirements that are currently required for echo cancellers."

3.3.2 Configuration and usage

3.3.2.1 Session configuration

There are many configuration possibilities planed.

It is possible to configure WinSip means the "preferences" dialog (WinSip → Options → Preferences) where the following components can be configured:

- Protocol: UDP or TCP
  For the time being only UDP is implemented.
- Local port
- SIP proxy
- Server port

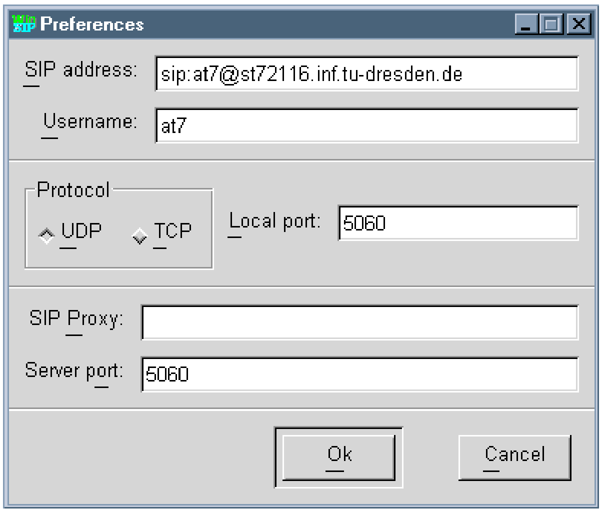
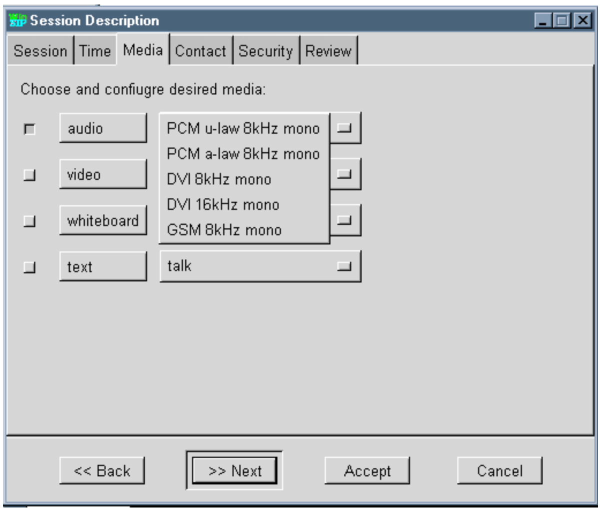
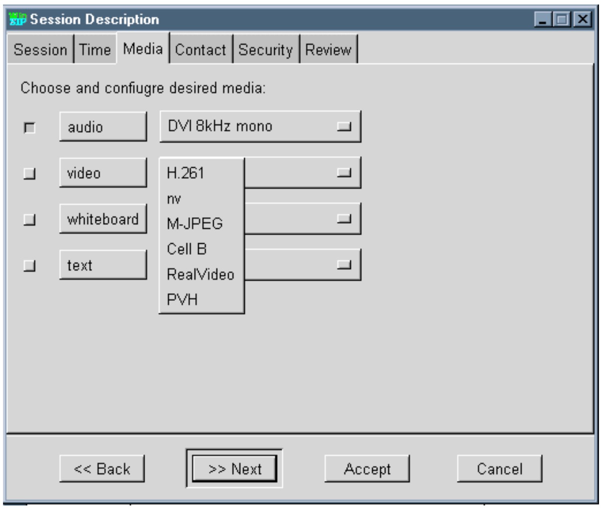


Figure 3-7: Configuration dialog of the preferences for a session

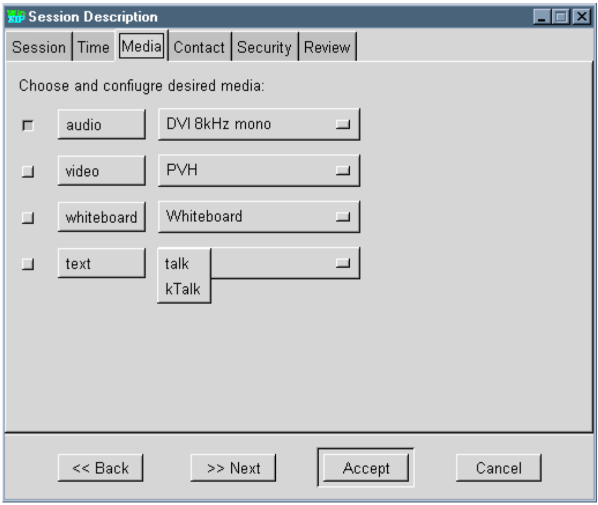
It is planed to configure the session by adjust the media components, and the contact component. For the moment only the "default" configuration is implemented.



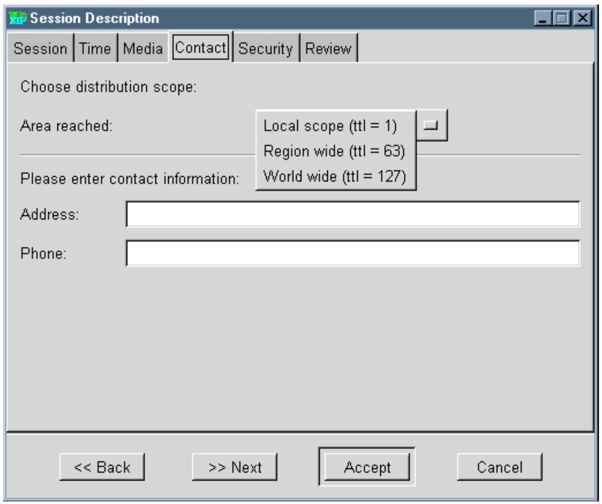
a. Media configuration: Audio



b. Media configuration: Video



c. Media configuration: Text



d. Contact configuration: Area reached

Figure 3-8: Session configuration: Media and contact

3.3.2.2 User guide

At the moment there is only one way to use WinSip.

- The preferences dialog should be set as in Figure 3-7.
- Every time that WinSip is opened it is necessary to create a new “default session”.
- Creation of buddies for the correspondents like in the following. The SIP address is required.

Example SIP address:

sip:ff1@st72113.inf.tu-dresden.de

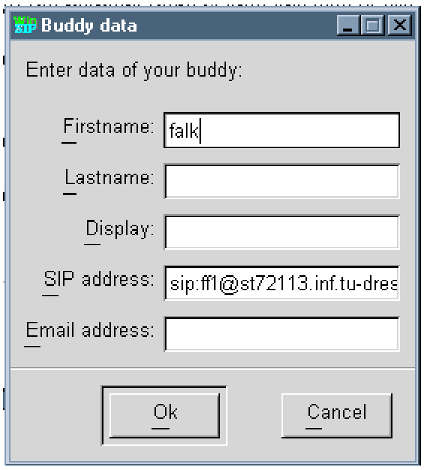


Figure 3-9: Buddy data: SIP address

- How to effectuate a call:
  1. Have a default session
  2. Select a buddy
  3. Click on the "call" icon, or click with the right mouse key on a "buddy" folder in the main window and select "call buddy"
  4. Select the last default session

3.3.3 Technical requirements

Application: Win SIP Client				
Version: 0.2.0				
Comments: WinSip is a Windows (©®™) application prototype which allows setup of full duplex telephone calls in a point-to-point fashion. The calls are signalled using SIP. The coded voice information is conveyed with RTP/UDP/IP. The available codec is PCM µ-Law in this version.				
Services components				
Audio				
Network characteristics				
Control port used by SIP: 5060; fix assigned (must be assigned in menu Options/Preferences)				
port used by RTP: 5004; fix assigned				
port used by RTCP: 5005; acc. RFC (RTP port +1, RTP port should be even)				
Technical characteristics				
built-in codec: PCM µ-Law (G.711)				
QoS specification				
(max) Delay in ms	Echo becomes a problem when the round-trip delay is more than 50 ms.		Talker overlap (the problem of one caller stepping on the other talker's speech) becomes significant if the one-way delay becomes greater than 250 ms.	
(max) Jitter in ms				
(max) Loss	Packet losses greater than 10% are generally not tolerable [Watson97]			
Ordering of packets				
BW guarantee	64 Kbps			
Traffic description				
Bandwidth	64 Kbps			
IP Packet size	554 bytes			
BSS in bytes				
(m)				
EAR	64 Kbps			
PR	67,5 Kbps			
SR	64 Kbps			
Possible traffic model				
Reservation style				
	p2p			
Possible network services				
	PCBR			

Table 3-4: WinSip application requirements

3.4 Multi-Player Online Games: Unreal Tournament

3.4.1 Description

Unreal Tournament is a 3D shooter where the player has to defend himself from clans. Many game modes are possible.

Note that we chose this game for its network characteristics and NOT for its rather non-ethical and violent content.

"Multi-player gaming is about shared reality: that all of the players feel they are in the same world, seeing from differing viewpoints the same events transpiring within that world.

Unreal introduces into multi-player gaming a new approach termed the generalised client-server model. In this model, the server is still authoritative over the evolution of the game state. However, the client actually maintains an accurate subset of the game state locally, and can predict the game flow by executing the same game code as the server, on approximately the same data, thus minimising the amount of data that must be exchanged between the two machines. Further, the 'game state' is self-described by an extensible, object-oriented scripting language which fully decouples the game logic from the network code." http://unreal.epicgames.com/

3.4.2 Significance of multi-player online games

- Multi-player games

Extract from [Nack01]:

*"If you look at the economic side of gaming, you can understand why the gaming industry drives digital entertainment. Datamonitor (*http://www.datamonitor.com/productdetail.asp?id=DMTC0704&ref=News%20Story*) estimates that the PC and console software sales in the US and Europe amounted to $10.9 billion in 2000. In another Datamonitor forecast on online gaming, the company stated that in 1999 8.4 million players world-wide played PC games online and that figure should reach 28 million by 2004. The forecast also points out that in 2004 we'll see an addition-al 48 million players using a console. This figure seems correct because Sony alone sold more than 70 million of its PlayStation 1 (without Internet access) consoles worldwide and it seems that PlayStation 2 (with Internet access) will become a similar success story. Although, that might depend on the Nintendo's forthcoming Dolphin (spring 2001) and Microsoft's X-Box (fall 2001), which promises near-photorealistic graphics and lifelike animation (*http://www.microsoft.com/presspass/features/2000/05-10bachqa.asp*). Thus, the game industry has shown tremendous performance in producing cutting-edge hardware and software technology. As researchers, we should closely watch what's happening in the field of games in particular and digital entertainment in general."*

- Paying to play

[Mehrotra00]:

"Nonetheless, the public is eager to get online and play – and they're willing to pay. A PC Data survey in June found that one third of home Internet users plan to buy one of the next-generation game consoles, which manufacturers promise will revolutionize online gaming. PlayStation 2 was the most desired of these new consoles, followed by Sega Dreamcast, Nintendo Dolphin, and Microsoft X-Box. Matt Gravett, games analyst for PC Data, qualified this

apparent consumer enthusiasm with a caution. 'Consumers are not expecting to pay more to game online via the next generation of consoles. Hardware prices and Internet access costs must be kept low, or else the online console gaming movement could stall.' On 26 October, Sony's PlayStation 2 hit the North American market at a cost of $299.99 per console. The first shipment of 500,000 units (which Sony characterized as a 'grave shortage') reportedly had sold out by 9:00 a.m. that morning. Sony expects to ship 100,000 more consoles a week through Christmas 2000.

What's at stake?

Although the current software for PlayStation 2 isn't geared toward online play, the PC Data survey suggests that the public is ready for console Internet access. Thus, online gaming is likely to expand tremendously along with the market it reaches. In addition, online gambling, though illegal in the US, is burgeoning, with Wall Street analysts expecting it to yield more than $5 billion a year in profits by 2003 (Los Angeles Times, 15 Nov. 2000)."

PC Data Study Shows Gamers postured for New Platforms: http://www.pcdataonline.com/press/pcdo060800.asp

3.4.3 Configuration and usage

Unreal tournament implements application level QoS adaptation in order to cope with the "best effort" packet delivery from the current Internet. The configuration possibilities are therefore quite rudimentary and aim to optimise the quality. The relevant configuration consist in setting the network type.

3.4.3.1 Session configuration

In the menu Option → Preferences the end-user can configure his session (video component, audio, game, network, ...). Only the network component configuration is relevant for QoS and can influence QoS.

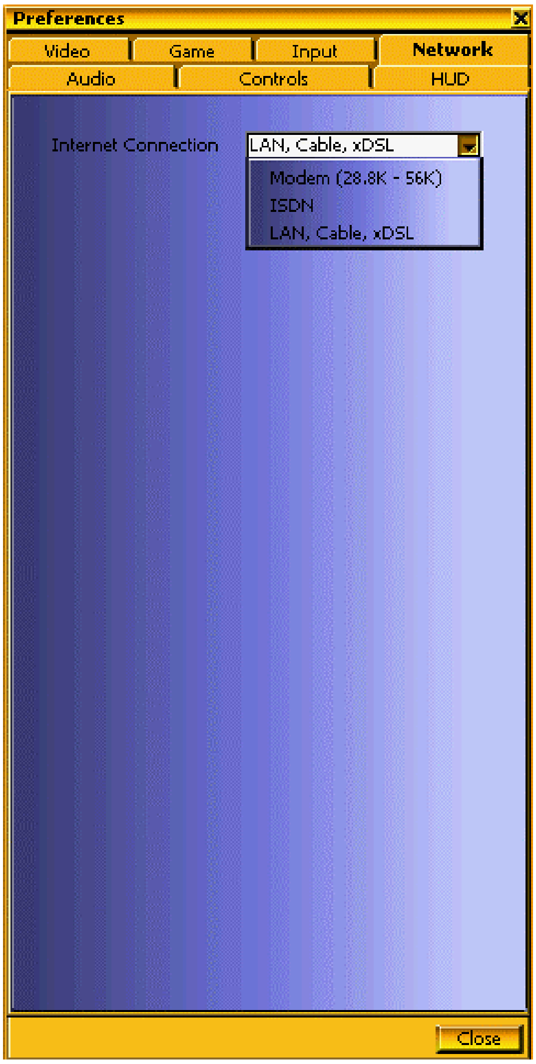


Figure 3-10: Screenshot of the network configuration

3.4.3.2 User Guide

Before beginning a game the end-user has to:

- configure his session,
- set up his "personality",
- select his weapons,

- choose the game server (LAN or WAN),

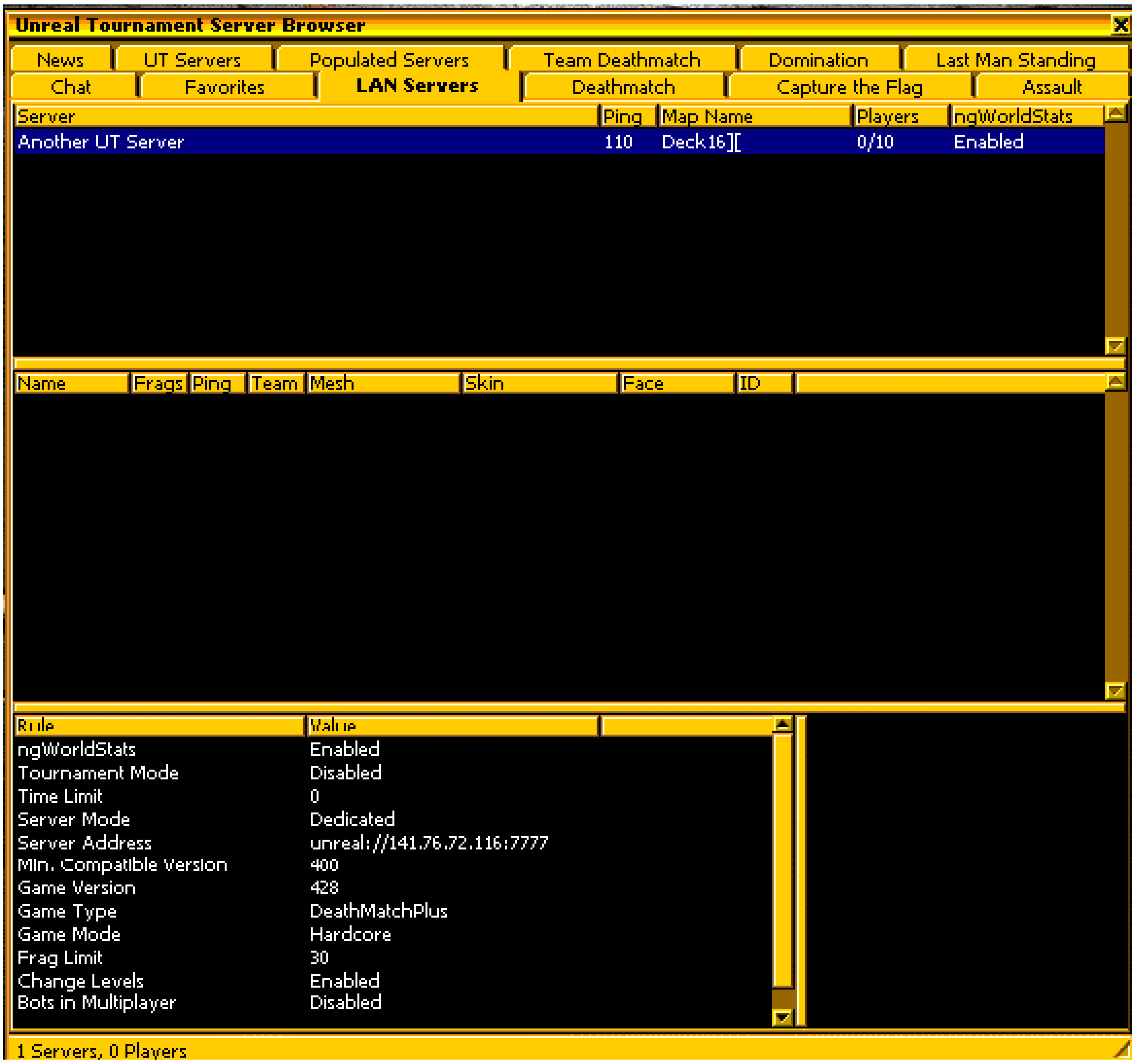


Figure 3-11: Screenshot of the server

- connect to chat (optional),
- start an online game,

- configure the game, and

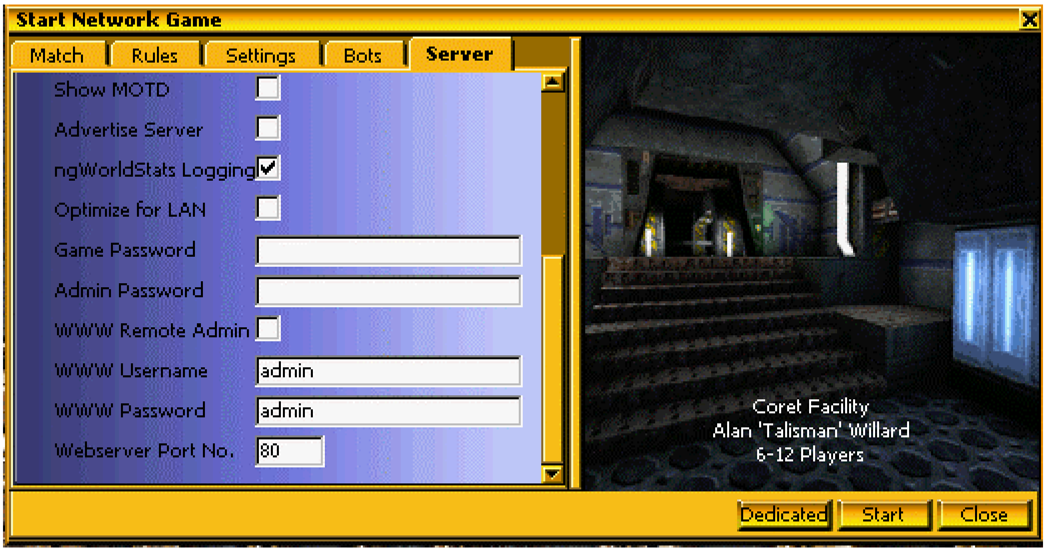


Figure 3-12: Screenshot of the game server settings

- play.

Game modes:

A player playing Unreal Tournament can choose one from the following game modes. The following information is taken from the homepage: http://www.unrealtournament.com/global/modes.html

  - Tournament Deathmatch / Tournament Team Deathmatch
  - Last Man Standing
  - Capture the Flag (CTF)
  - Domination
  - Assault
  - Chaos

- Controls (information taken from the home page)
  - Keyboard

You can bind new keys to controls by clicking on the “Controls” tab of the Preferences menu inside of Unreal Tournament. There are a lot of controls to bind, so here is a list of defaults.

Key	Function
TAB	Opens a mini system console line.
Shift	Hold shift while moving to walk slowly.
V	Press and hold V to access the speech menu. Use this menu to give orders and taunt enemies.
F	Drop to the ground and pretend you are dead. (Feign death.)
~	Open the system command console.
R	Send a chat message to your team only channel.
T	Send a chat message to the public channel.
F11	Adjust screen gamma.
F9	Take a screenshot. (Appears in your UnrealTournament\System directory.)
F6, F7	QuickSave and QuickLoad respectively.
F5	Allows you to cycle through your teammates viewpoints.
F3	In an Assault game, F3 lists the current objectives.
F2	Displays the message of the day.
F1	Displays the scoreboard
H, J, K, L	These keys will cause your character to animate in a taunt.
Number Keys [0..9]	The number keys cycle through your weapons in order.
Arrow Keys	The arrow keys will move you forward and back or strafe left and right.
Escape	The Escape key will open the menus.
Pause	Pause will freeze the game.
Alt	Alt will fire your weapon’s primary attack.
Ctrl, Space	Ctrl or Space causes your player to jump.

○ Mouse

Although you can rely solely on your keyboard to move around in and interact in Unreal Tournament’s 3D universe, using both the keyboard and mouse simultaneously gives you much more fluid and responsive control.

When you use the mouse to control your rotational movement and aiming you gain a degree of precision and speed that players using keyboard-only controls cannot touch. The keyboard is best used for easy lateral and forward/backward movement, and for jumping.

To master the default controls in Unreal Tournament, keep your left hand on the keyboard, using the arrow keys for movement, the 0-9 keys for weapon selection, and the space bar for jumping. Your right hand operates the mouse, controlling rotation, aiming, and firing. Of course, you can customise these controls to suit your preferences via the Options Menu.

- Speech Menu

Press and hold the V key to open the speech menu. While you hold V your mouse will become activated and you can select various commands and taunts. Under the Orders submenu you can select a job to assign to a team-mate. If your bot is a team-mate they will automatically carry out your orders.

The available orders are:

Defend the Base.	The ordered bot will immediately make his way to your base and protect it from enemy attack.
Cover Me.	The bot will find you and follow you attacking enemies that you encounter.
Assault the Base.	Orders the bot to go offensive. In CTF this order is replaced with “Capture the Enemy Flag.”
Hold this Position.	The bot will find your current location and try to protect it from enemy attack.
Freelance.	Releases the bot from previous orders. A freelancing bot will make their own battlefield decisions.

If you give an order to “All” then every bot on your team will attempt to complete the order. If you look at a bot the speech menu will have the option to “Order This Bot.”

3.4.4 Technical requirements client

Application: Unreal Tournament			
Version:			
Comments: The configuration of the game is such that there is a single server and a number of clients which communicate with the server. Most of the things are done in the clients, but the movement and shooting of players is reported to the server, which also provides relevant info on the movement & shooting of other players and "monsters" and the placement of movable objects.			
Services components			
Data			
Network characteristics			
UDP			
Port: 7777 (server port can be set)			
Net speed : "*If you are seeing significant lag while playing Unreal Tournament, try adjusting your netspeed. You can do this by typing netspeed xxxx (where xxxx is a value in bytes/sec). Net speed is set by default based on your selection of your network connection in the networking menu. The default values are 2600 for modems, 5000 for ISDN, and 20000 for xDSL, cable modem, and LAN. For example, some cable modems limit upstream bandwidth. For cable modems, try netspeed 10000 or lower if you are seeing poor network performance.*"			
28.8 K modem connection at least needed			
Technical characteristics			
e.g. codecs used			
QoS specification			
(max) Delay in ms	RTT: 100 ms is noticeable	RTT: 150 ms harmful	RTT: 200 ms play is diffi-cult
(max) Jitter in ms			
(max) Loss	5% (official info)		
Ordering of packets	important		
BW guarantee	28.8 K?		
Traffic description			
Session Scenario	LAN, xDSL, cable modem	ISDN	Modem
Bandwidth	30 packets/s 20 Kbps	5 Kbps	2,6 Kbps
Packet size	50-65 Bytes		
BSS in bytes			
(m)			
EAR			
PR in bit*s^{-1}			
SR in bit*s^{-1}			
Possible traffic model			
Reservation style			
	p2p (from the client)		
Possible network services			
	PMC		

Table 3-5: Unreal Tournament client requirements

Unreal may be a good application for PMC, since the bandwidth required for playing the game on a network is relatively small, a low delay influences the game, and a small packet loss (5%) can have effects on the playability of this game.

3.4.5 Technical requirements server

Application: Unreal Tournament			
Version:			
Comments: The configuration of the game is such that there is a single server and a number of clients which communicate with the server. Most of the things are done in the clients, but the movement and shooting of players is reported to the server, which also provides relevant info on the movement & shooting of other players and "monsters" and the placement of movable objects. Make sure to match the maximum number of users for your server to the bandwidth you have available. The amount of bandwidth required for each user is determined by the MaxClientRate, which is set by default to 20000 bytes/sec. If your upstream bandwidth is limited (for example a cable modem), you should reduce this value. Make sure that the number of players your server allows (MaxPlayers) multiplied by MaxClientRate is less than your available bandwidth (in the case of cable modems or xDSL, your upstream bandwidth).			
Services components			
Data			
Network characteristics			
UDP			
Port: 7777 (server port can be set)			
Technical characteristics			
e.g. codecs used			
QoS specification			
(max) Delay in ms	RTT: 100 ms is noticeable	RTT: 150 ms harmful	RTT: 200 ms play is difficult
(max) Jitter in ms			
(max) Loss	5% (official info)		
Ordering of packets	important		
BW guarantee	28.8 K?		
Traffic description			
Bandwidth	30 packets/s	30 Kbps	
Packet size	90-180 bytes		
BSS in bytes			
(m)			
EAR			
PR in bit*s^{-1}			
SR in bit*s^{-1}			
Possible traffic model			
Reservation style			
	p2a (from the server)		
Possible network services			
	PMC		

Table 3-6: Unreal Tournament server requirements

3.5 Multi-Player Online Games: Ultima Online

3.5.1 Description

This game is designed and developed by Origin Systems Inc. / Houston, Texas USA. The game resembles medieval age Role Playing Atmosphere, by utilising an isometric view.

In a role playing game the player can live in a fantasy world with mystic figures, magicians, monsters, etc. Performing one or more fictive characters (roles) such as the fighter, the sorcerer, the thief, etc. The special feature of online role playing games is that the human players can interact anonymously with each other.

Furthermore Ultima Online offers a number of special features (http://www.uo.com), such as:

- The players can customise their characters in 45 skills.
- There are 8 different circles of magic with 64 different magical spells.
- The game offers the players unique opportunity to buy homes, buildings, boats and even castles.
- The main world “Britannia” has 13 major cities across the continent.
- There are many different beasts and monsters like alligators, zombies, etc.

With these features and the interaction between the players a fantasy world can be created offering a very strong feeling of “reality”, ranging from safe cities with guards and people all over to woods and dungeons crawling with beasts and high level monsters.

Figure 3-13: Screenshot of Ultima Online

3.5.2 Connection problems

Being TCP based, the game requires answers from the server in order to continue. The player is allowed to walk up to three steps. If during this time the client does not get an answer from the server about the current state of the surrounding game play area, the game "freezes" or halts until this answer is received. If the client freezes, the game experience can be affected in different ways, depending on the current situation.

Loss of communication for longer than 5 seconds:

- If the player is in a situation considered as "safe", for instance being in a town, in a guard-house talking to another player character, then the "frozen" status just leads to the conversation being interrupted. Unless the client looses connection to the server for longer than 30 seconds resulting in a client/server disconnect and requiring the player to log in to the game again, this situation is acceptable although the majority of the gaming "community" do not really like it.

- If the player is in a situation considered as “unfriendly”, for instance in a dungeon close to a creature able to kill the player within seconds, packet loss is close to unacceptable. The player could loose his/her “life” resulting in often tedious procedures to regain the status where the player has to start battling the creature.

Small amounts of packet-loss and / or ping-times above 400 ms:

This results in the condition where the player walks significantly slower than the others do in this game. While the server notes that the client of others had been send an answer therefore allowing them to walk again, the player has to wait to receive a answer from the server. The game play window still progresses, but everyone and everything around the player is able to “outrun” and therefore play the game faster as the player. The player is unable to battle any creature, but he undertakes to complete tasks such as cutting wood in the forest or mining iron in the mountains, fishing, cooking and everything not interfering with hostile creatures. This enables the player to “live” in the game even with a certain amount of packet loss or a bad ping-time and therefore bad connection.

3.5.3 Technical requirements

Application: Ultima Online				
Version:				
Comments: One or more servers build up the role playing world of Ultima Online. A number of clients communicate with the server(s). For testing only one server is needed. Most of the things are done in the clients, but the movement and interaction of players is reported to the server, which also provides relevant information on the movement and interaction of the other human players and computer players, monsters, objects, etc. Free server LINUX: http://www.sphereserver.com Free server Windows and game world files: http://www.uoxdev.com For server and client the original game of Ultima Online is necessary. To start the game call the ignition tool in the Ultima folder. For installation information it is necessary to read the readme file located in the same folder. More information will be found under "uoxdev.com"				
Services components				
Data				
Network characteristics				
TCP				
Port: 2593				
28.8K modem connection at least needed				
Technical characteristics				
QoS specification				
(max) Delay in ms	< 150 ms the game runs smooth	150 – 300 ms are noticeable	300 – 400 ms the game is difficult to play	> 400 ms the game is unplayable
(max) Jitter in ms				
(max) Loss	< 2% the game runs smooth	> 2 % the game is unplayable		
Ordering of packets	important			
BW guarantee	28.8 Kbps (ping < 200ms)	64 Kbps		
Traffic description				
Bandwidth	2 kbyte/s			
Packet size				
BSS in bytes				
(m)				
EAR				
PR in bit*s^{-1}				
SR in bit*s^{-1}				
Possible traffic model				
Reservation style				
	p2p (from the client) p2a (from the servre)			
Possible network services				
	PMC			

Table 3-7: Ultima Online requirements

Ultima Online may be a good application for PMC, since the bandwidth required for playing the game is relatively small, a low delay possibly influences the game, and a small packet loss (>2%) can have effects on the playability of this game.

4 Application Profiles for Legacy Applications

4.0 Approach: refinement of [D2201, chap. 4.2]

4.0.1 Structure of the application profile

An application can be described at two different levels of abstraction. The end-user (e.g. player of a multi-player online game) has a "human" oriented view of the application, whereas the network only uses / needs technical parameters.

Extract from [Eichler01]: *"At the application level, the end-user chooses QoS with a verbal description corresponding to a universal apprehension of applications. Considering the sample case of a video-conferencing session, the user can formulate its QoS requests by selecting from a set of proposals the size of the picture, the frame rate, the sound quality etc. These parameters relevant to the application level are referred to as session characteristics.*

At the network level, a QoS request represents a list of network specific parameters related to low-level traffic and QoS parameters such as peak rate and jitter. All those parameters are referred to as technical characteristics, and are included in the QoS request message which is sent by the EAT to the ACA, along with the specification of the selected NS."

The profiles and their parameters build one part of the persistence layer used by the EAT.

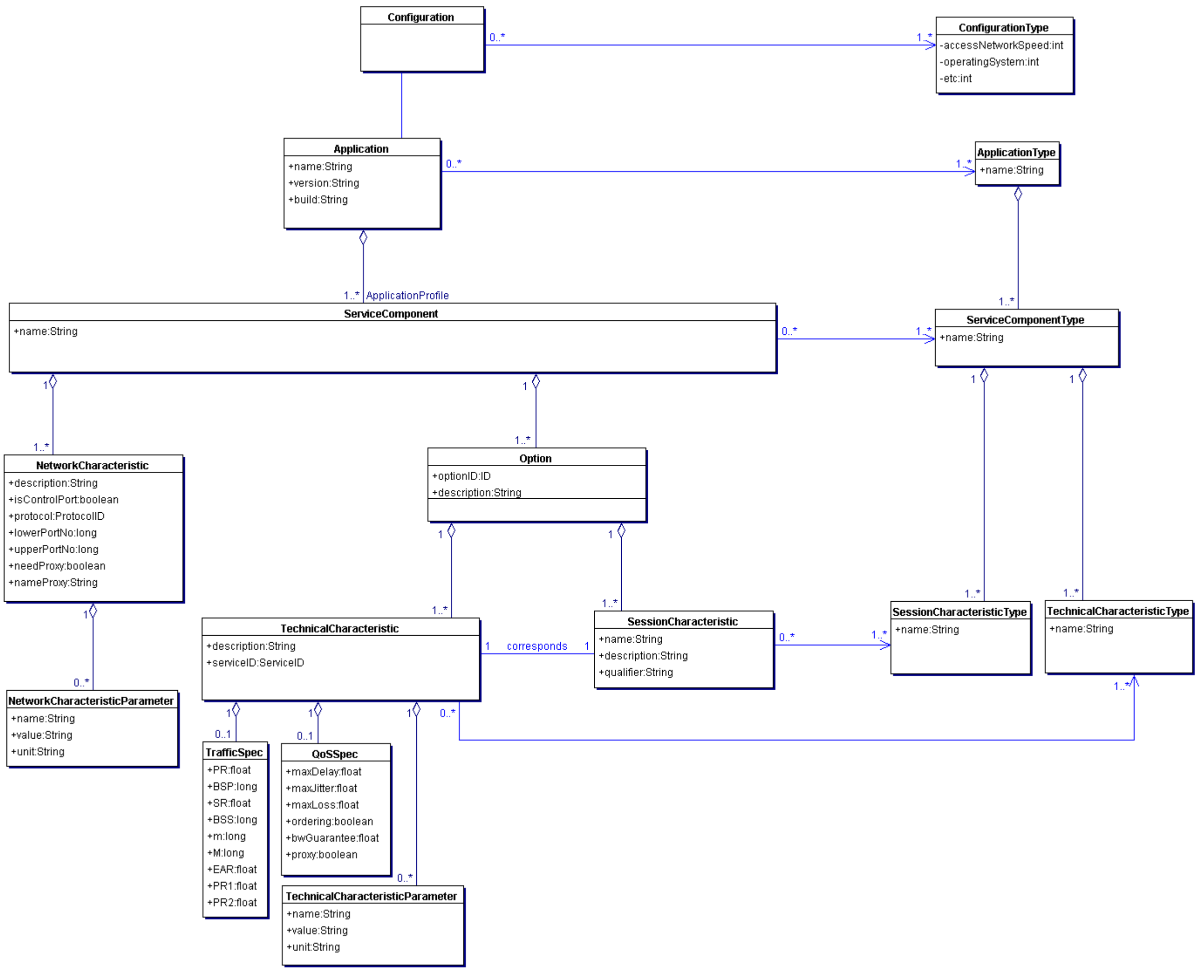


Figure 4-1: Class diagram of the Application Profile structure

4.0.2 Analysis patterns in the application profile structure

Extract from [Eichler01]: *"Applications can be described from a general point of view and from a concrete one. First, at a general level, an* `Application` *can be associated to* `ApplicationTypes` *like 'video conferencing', 'voice over IP', 'streaming media', etc. The 'video conferencing'* `ApplicationType` *for example is described by the* `ServiceComponentTypes:` *'video', 'voice' and 'data'. A* `ServiceComponentType` *itself is described on the one hand by network-oriented* `TechnicalCharacteristicTypes` *e.g. the codecs, on the other hand by the user-oriented* `SessionCharacteristicTypes` *like 'picture size', 'sound quality', etc. The slightly modified item-item description analysis pattern [Ambler98] (...) allows the modelling of such relations between objects and shared descriptions where an item object is described by a description item. In our case: a* `ServiceComponentType` *(the description item) describes an* `ApplicationType` *(the item object).* `SessionCharacteristicTypes` *and* `TechnicalCharacteristicTypes` *describe a* `ServiceComponentType`*, etc.*

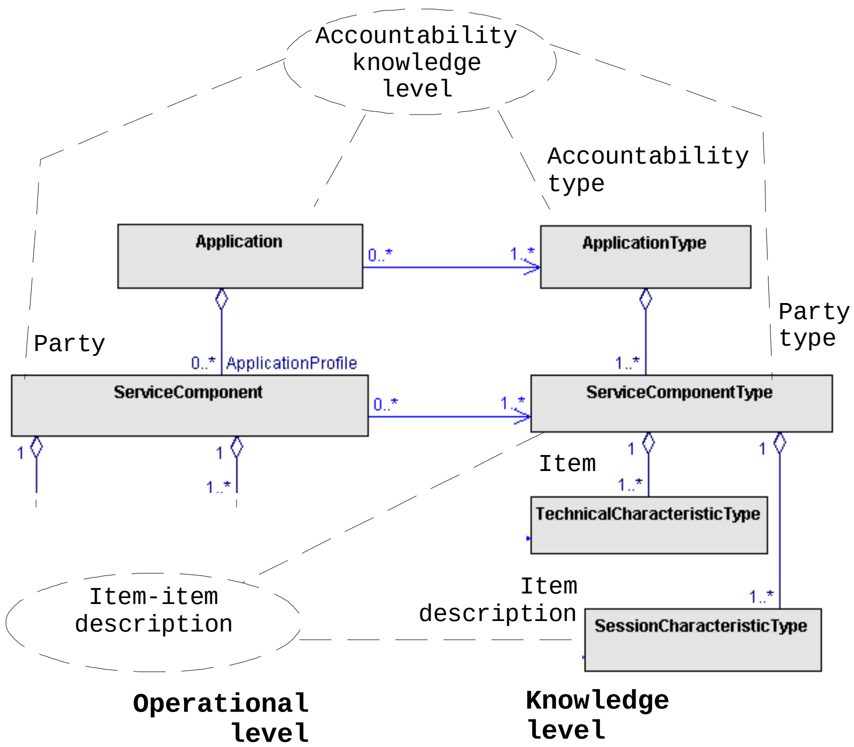


Figure 4-2: Application Profile class diagram. The adapted item-item description analysis pattern and the adapted accountability knowledge level analysis pattern are depicted using UML.

Second, the analysis of concrete applications leads to the following statement. A concrete application like NetMeeting from Microsoft shares with similar applications the same 'video conferencing' `ApplicationType`*. As previously seen, such an* `ApplicationType` *can be modelled at a general level. The structure describing the concrete application (e.g. NetMeeting) should follow the same description rules as the one of the* `ApplicationType`*. The adapted accountability knowledge level analysis pattern [Fowler98] (...), allows the split of the overall model in two sections: an operational one (corresponding to the concrete application), and a knowledge one (corresponding to the application type). At the knowledge level the model records the general rules that govern the structure and corresponds to a meta-level description of an application profile."*

4.0.3 Realisation of the profiles

- Persistence problem:
    - We will support an arbitrary amount of generic application types.
    - To each application type corresponds an arbitrary amount of session characteristics.

  - To each application type corresponds an arbitrary amount of technical characteristics.
  - To each session characteristic corresponds an arbitrary amount of qualifiers.
  - To each technical characteristic corresponds an arbitrary amount of parameters.
- Presentation problem for the end-user in a session:
  - For each application profile the converter calculates an arbitrary amount of session characteristic sets.
  - To each session characteristic set corresponds (depending of the application type of the application) an arbitrary amount of session characteristics with qualifier.
- Hierarchical structure of the profile: As depicted the structure of an application profile is hierarchical.

4.0.3.1 Solution proposal

In this section we give very briefly one possible development solution. (For further technical options, please refer to [D2201].)

Dynamic Web application:

- For personalisation:

  Allowing the end-user to configure the information he sees

  Remembering the end-user preferences / purchases

- For manageability:

  Have the content independent from the layout and the logic

- Integration:

  Interfacing web-based request to back-end systems

- Existing technologies:

  For the state of the art analysis please refer to [D2201, chap. 6]. The Figure 4-3 shows a general view of different Java technologies and their relations for a Web application.

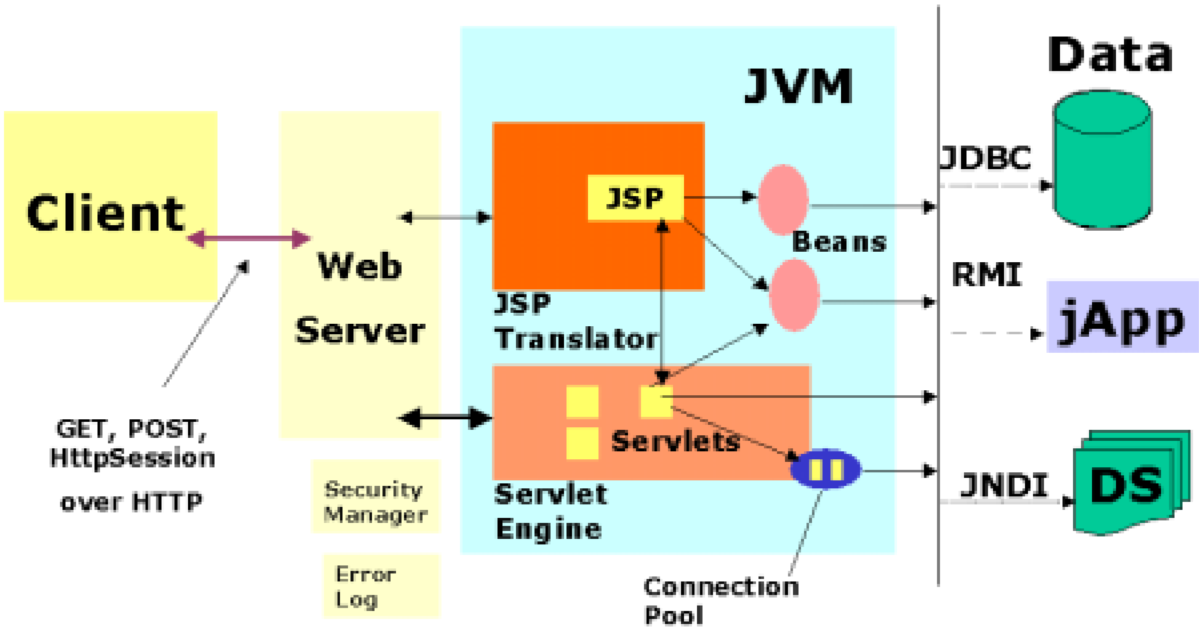


Figure 4-3: Java Web application and the different technologies

- Web application with server sided logic: The EAT is working in a Web browser.
    - JavaServer Pages
    - HTML / XML for the GUI
    - Servlet / JavaBeans for the logic
    - In a first step XML as persistence mechanism for the application profiles. Latter XML could be used as an exchange format between a database and the EAT.
    - DTDs / XML to describe the application profiles
    - Architecture proposal for the EAT as a Web application:

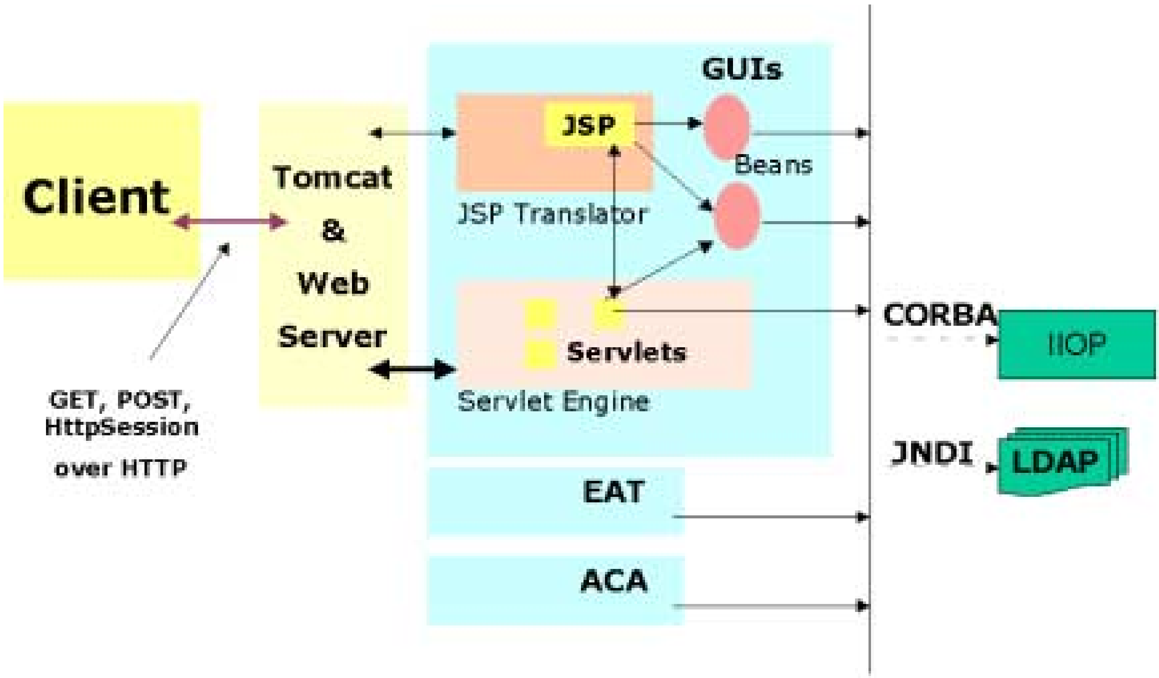


Figure 4-4: Architecture proposal for the EAT

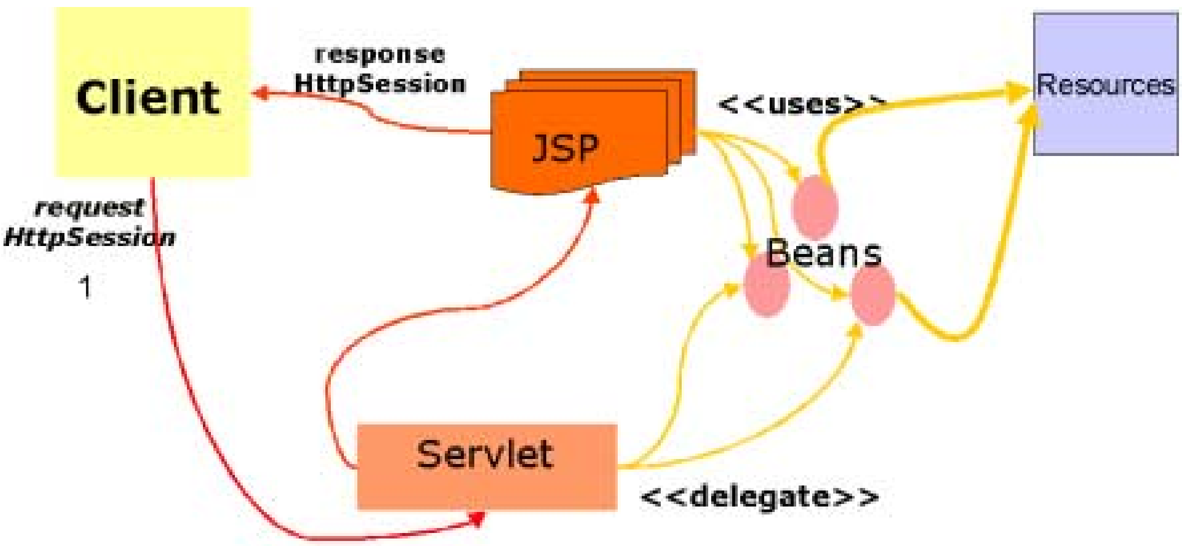


Figure 4-5: Relations between the different elements, and "calling" proposal

4.0.4 Refined application profile as XML / DTD

The class diagram representing the application profile has been transcribed in a DTD file (the schema of the XML file, Figure 4-7), and then in an XML file (Figure 4-8). First, the profile graph is depicted in Figure 4-6.

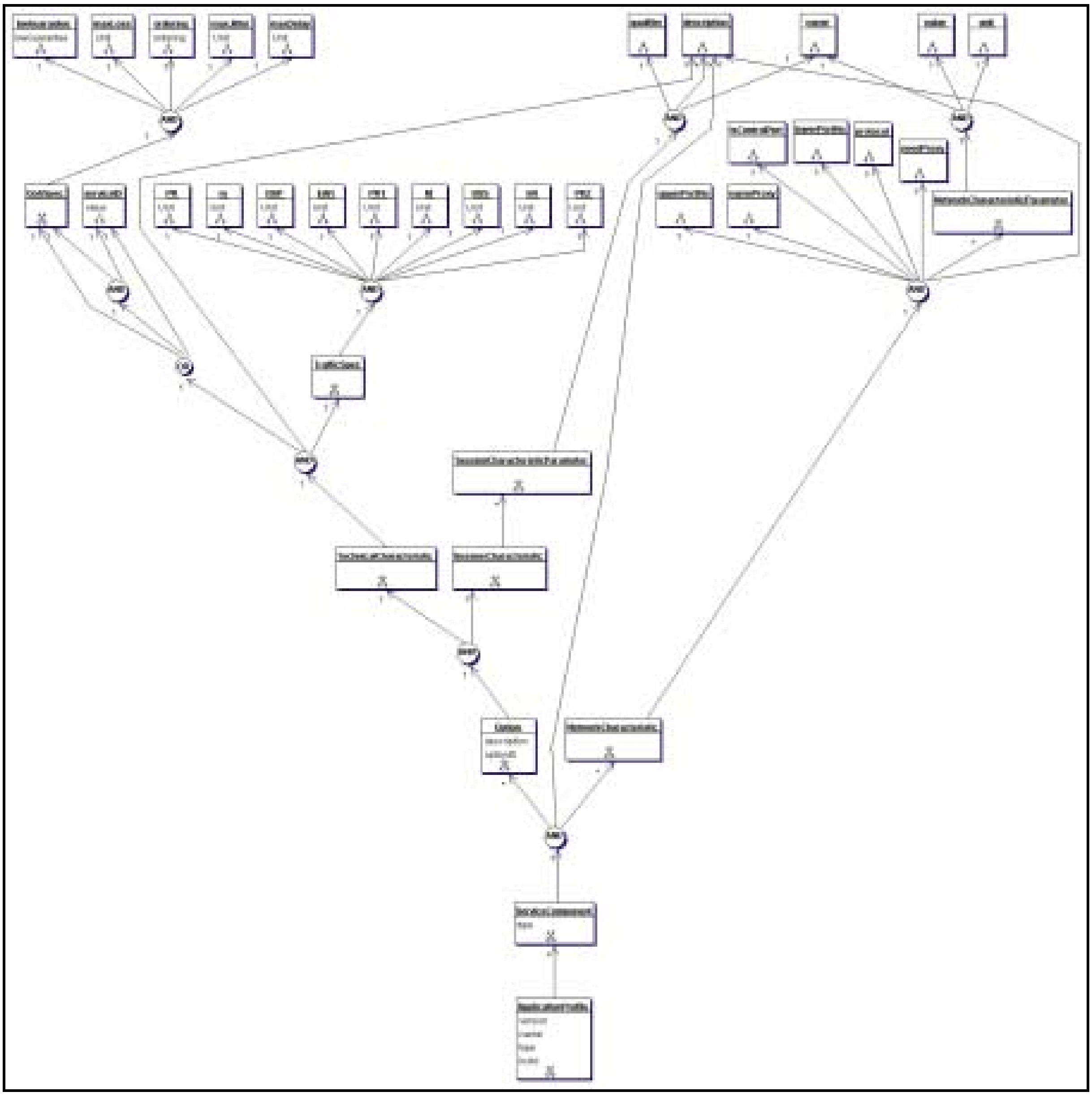


Figure 4-6: Application Profile graph

ApplicationProfileV6.dtd:

```
<!ELEMENT ApplicationProfile (ServiceComponent+)>
<!ATTLIST ApplicationProfile
    name CDATA #REQUIRED
    version CDATA #REQUIRED
    build CDATA #IMPLIED
```

```
    type (VoIP | GAME | MULTIMEDIA | STREAMINGVIDEO | STREAMINGAUDIO) #REQUIRED
>
<!ELEMENT ServiceComponent (description?, NetworkCharacteristic+, Option+)>
<!ATTLIST ServiceComponent
    type (AUDIO | VIDEO | DATA) #REQUIRED
>
<!ELEMENT description (#PCDATA)>
<!ELEMENT NetworkCharacteristic (description?, needProxy, nameProxy?, lowerPortNo?,
upperPortNo?, isControlPort?, protocol?, NetworkCharacteristicParameter*)>
<!ELEMENT Option (SessionCharacteristic, TechnicalCharacteristic)>
<!ATTLIST Option
    optionID CDATA #REQUIRED
    description CDATA #IMPLIED
>
<!ELEMENT SessionCharacteristic (SessionCharacteristicParameter+)>
<!ELEMENT TechnicalCharacteristic (description?,((serviceID |
QoSSpec)|(serviceID,QoSSpec)), TrafficSpec)>
<!ELEMENT needProxy (#PCDATA)>
<!ELEMENT nameProxy (#PCDATA)>
<!ELEMENT lowerPortNo (#PCDATA)>
<!ELEMENT upperPortNo (#PCDATA)>
<!ELEMENT isControlPort (#PCDATA)>
<!ELEMENT protocol (#PCDATA)>
<!ELEMENT NetworkCharacteristicParameter (name, value?, unit?)>
<!ELEMENT SessionCharacteristicParameter (name, description, qualifier)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT qualifier (#PCDATA)>
<!ELEMENT serviceID EMPTY>
<!ATTLIST serviceID
    value (PCBR | PVBR | PMM | PMC | STD | CUSTOM) "STD"
>
<!ELEMENT QoSSpec (maxDelay, maxJitter, maxLoss, bwGuarantee, ordering)>
<!ELEMENT TrafficSpec (PR, BSP, SR, BSS, m, M, EAR, PR1, PR2)>
<!ELEMENT value (#PCDATA)>
<!ELEMENT unit (#PCDATA)>
<!ELEMENT maxDelay (#PCDATA)>
<!ATTLIST maxDelay
    Unit CDATA #FIXED "ms"
>
<!ELEMENT maxJitter (#PCDATA)>
<!ATTLIST maxJitter
    Unit CDATA #FIXED "ms"
>
<!ELEMENT maxLoss (#PCDATA)>
<!ATTLIST maxLoss
    Unit CDATA #FIXED "float"
>
<!ELEMENT bwGuarantee (#PCDATA)>
<!ATTLIST bwGuarantee
    bwGuarantee CDATA #FIXED "kb/s"
>
<!ELEMENT ordering (#PCDATA)>
<!ATTLIST ordering
    ordering CDATA #FIXED "boolean"
>
<!ELEMENT PR (#PCDATA)>
<!ATTLIST PR
    Unit CDATA #FIXED "bit/s"
>
<!ELEMENT BSP (#PCDATA)>
<!ATTLIST BSP
    Unit CDATA #FIXED "bytes"
>
<!ELEMENT SR (#PCDATA)>
<!ATTLIST SR
    Unit CDATA #FIXED "bit/s"
>
```

```
<!ELEMENT BSS (#PCDATA)>
<!ATTLIST BSS
    Unit CDATA #FIXED "bytes"
>
<!ELEMENT m (#PCDATA)>
<!ATTLIST m
    Unit CDATA #FIXED "bytes"
>
<!ELEMENT M (#PCDATA)>
<!ATTLIST M
    Unit CDATA #FIXED "bytes"
>
<!ELEMENT EAR (#PCDATA)>
<!ATTLIST EAR
    Unit CDATA #FIXED "bit/s"
>
<!ELEMENT PR1 (#PCDATA)>
<!ATTLIST PR1
    Unit CDATA #FIXED "bit/s"
>
<!ELEMENT PR2 (#PCDATA)>
<!ATTLIST PR2
    Unit CDATA #FIXED "bit/s"
>
```

Figure 4-7: Application Profile DTD file

ApplicationProfileV6.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ApplicationProfile SYSTEM "ApplicationProfileV6.dtd">
<ApplicationProfile name="value" version="value" build="value" type="VoIP">
    <ServiceComponent type="AUDIO">
        <description></description>
        <NetworkCharacteristic>
            <description></description>
            <needProxy></needProxy>
            <nameProxy></nameProxy>
            <lowerPortNo></lowerPortNo>
            <upperPortNo></upperPortNo>
            <isControlPort></isControlPort>
            <protocol></protocol>
            <NetworkCharacteristicParameter>
                <name></name>
                <value></value>
                <unit></unit>
            </NetworkCharacteristicParameter>
        </NetworkCharacteristic>
        <Option optionID="value" description="value">
            <SessionCharacteristic>
                <SessionCharacteristicParameter>
                    <name></name>
                    <description></description>
                    <qualifier></qualifier>
                </SessionCharacteristicParameter>
            </SessionCharacteristic>
            <TechnicalCharacteristic>
                <description></description>
                <serviceID value="STD"/>
                <QoSSpec>
                    <maxDelay Unit="ms"></maxDelay>
                    <maxJitter Unit="ms"/>
                    <maxLoss Unit="float"></maxLoss>
                    <bwGuarantee bwGuarantee="kb/s"></bwGuarantee>
                    <ordering ordering="boolean"></ordering>
                </QoSSpec>
```

```
                <TrafficSpec>
                    <PR Unit="bit/s"></PR>
                    <BSP Unit="bytes"></BSP>
                    <SR Unit="bit/s"></SR>
                    <BSS Unit="bytes"></BSS>
                    <m Unit="bytes"></m>
                    <M Unit="bytes"></M>
                    <EAR Unit="bit/s"></EAR>
                    <PR1 Unit="bit/s"></PR1>
                    <PR2 Unit="bit/s"></PR2>
                </TrafficSpec>
            </TechnicalCharacteristic>
        </Option>
    </ServiceComponent>
</ApplicationProfile>
```

Figure 4-8: Application Profile XML file

The further steps are the creation of the XML profiles by using the values of the concrete application requirements.

For some more information on DTDs and XML please refer to document IST-1999-10077-WP2.2-TUD-00034-TC-CC/b.)

4.1 Profile for RealSystem

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Andreas Koenig (BAG) -->
<!DOCTYPE ApplicationProfile SYSTEM "ApplicationProfileV6.dtd">
<ApplicationProfile name="RealSystem" version="7.0" type="STREAMINGVIDEO">
    <ServiceComponent type="VIDEO">
        <description>Video with audio subdivided into five mains connection
categories</description>
        <NetworkCharacteristic>
            <description/>
            <needProxy>No</needProxy>
            <nameProxy/>
            <lowerPortNo>554</lowerPortNo>
            <upperPortNo/>
            <isControlPort>No</isControlPort>
            <protocol>TCP</protocol>
        </NetworkCharacteristic>
        <NetworkCharacteristic>
            <description/>
            <needProxy>No</needProxy>
            <nameProxy/>
            <lowerPortNo>6970</lowerPortNo>
            <upperPortNo/>
            <isControlPort>No</isControlPort>
            <protocol>UDP</protocol>
        </NetworkCharacteristic>
        <Option optionID="1" description="Modem">
            <SessionCharacteristic>
                <SessionCharacteristicParameter>
                    <name>Video quality</name>
                    <description>Very low quality video</description>
                    <qualifier>"very low"</qualifier>
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Window size</name>
                    <description>Window size</description>
                    <qualifier>"very small"</qualifier>
                </SessionCharacteristicParameter>
```

```
                <SessionCharacteristicParameter>
                    <name>Network Speed</name>
                    <description>Modem</description>
                    <qualifier>"very slow"</qualifier>
                </SessionCharacteristicParameter>
            </SessionCharacteristic>
            <TechnicalCharacteristic>
                <description/>
                <serviceID value="PMM"/>
                <TrafficSpec>
                    <PR Unit="bit/s">28800</PR>
                    <BSP Unit="bytes">360</BSP>
                    <SR Unit="bit/s"/>
                    <BSS Unit="bytes"/>
                    <m Unit="bytes"/>
                    <M Unit="bytes"/>
                    <EAR Unit="bit/s"/>
                    <PR1 Unit="bit/s"/>
                    <PR2 Unit="bit/s"/>
                </TrafficSpec>
            </TechnicalCharacteristic>
        </Option>
        <Option optionID="2" description="Single ISDN">
            <SessionCharacteristic>
                <SessionCharacteristicParameter>
                    <name>Video quality</name>
                    <description>Low quality video</description>
                    <qualifier>"low"</qualifier>
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Window size</name>
                    <description>Window size</description>
                    <qualifier>"small"</qualifier>
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Network Speed</name>
                    <description>Single ISDN</description>
                    <qualifier>"slow"</qualifier>
                </SessionCharacteristicParameter>
            </SessionCharacteristic>
            <TechnicalCharacteristic>
                <description/>
                <serviceID value="PMM"/>
                <TrafficSpec>
                    <PR Unit="bit/s">64000</PR>
                    <BSP Unit="bytes">421</BSP>
                    <SR Unit="bit/s"/>
                    <BSS Unit="bytes"/>
                    <m Unit="bytes"/>
                    <M Unit="bytes"/>
                    <EAR Unit="bit/s"/>
                    <PR1 Unit="bit/s"/>
                    <PR2 Unit="bit/s"/>
                </TrafficSpec>
            </TechnicalCharacteristic>
        </Option>
        <Option optionID="3" description="Dual ISDN">
            <SessionCharacteristic>
                <SessionCharacteristicParameter>
                    <name>Video quality</name>
                    <description>Medium quality video</description>
                    <qualifier>"good"</qualifier>
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Window size</name>
                    <description>Window size</description>
                    <qualifier>"medium"</qualifier>
```

```
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Network Speed</name>
                    <description>Dual ISDN</description>
                    <qualifier>"medium"</qualifier>
                </SessionCharacteristicParameter>
            </SessionCharacteristic>
            <TechnicalCharacteristic>
                <description/>
                <serviceID value="PMM"/>
                <TrafficSpec>
                    <PR Unit="bit/s">128000</PR>
                    <BSP Unit="bytes">488</BSP>
                    <SR Unit="bit/s"/>
                    <BSS Unit="bytes"/>
                    <m Unit="bytes"/>
                    <M Unit="bytes"/>
                    <EAR Unit="bit/s"/>
                    <PR1 Unit="bit/s"/>
                    <PR2 Unit="bit/s"/>
                </TrafficSpec>
            </TechnicalCharacteristic>
        </Option>
        <Option optionID="4" description="Cabel/DSL">
            <SessionCharacteristic>
                <SessionCharacteristicParameter>
                    <name>Video quality</name>
                    <description>High quality video</description>
                    <qualifier>"high"</qualifier>
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Window size</name>
                    <description>Window size</description>
                    <qualifier>"large"</qualifier>
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Network Speed</name>
                    <description>Cabel/DSL</description>
                    <qualifier>"fast"</qualifier>
                </SessionCharacteristicParameter>
            </SessionCharacteristic>
            <TechnicalCharacteristic>
                <description/>
                <serviceID value="PMM"/>
                <TrafficSpec>
                    <PR Unit="bit/s">256000</PR>
                    <BSP Unit="bytes">751</BSP>
                    <SR Unit="bit/s"/>
                    <BSS Unit="bytes"/>
                    <m Unit="bytes"/>
                    <M Unit="bytes"/>
                    <EAR Unit="bit/s"/>
                    <PR1 Unit="bit/s"/>
                    <PR2 Unit="bit/s"/>
                </TrafficSpec>
            </TechnicalCharacteristic>
        </Option>
        <Option optionID="5" description="LAN">
            <SessionCharacteristic>
                <SessionCharacteristicParameter>
                    <name>Video quality</name>
                    <description>Very high quality video</description>
                    <qualifier>"very high"</qualifier>
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Window size</name>
                    <description>Window size</description>
```

```
                    <qualifier>"very large"</qualifier>
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Network Speed</name>
                    <description>LAN</description>
                    <qualifier>"very fast"</qualifier>
                </SessionCharacteristicParameter>
            </SessionCharacteristic>
            <TechnicalCharacteristic>
                <description/>
                <serviceID value="PMM"/>
                <TrafficSpec>
                    <PR Unit="bit/s">1000000</PR>
                    <BSP Unit="bytes">755</BSP>
                    <SR Unit="bit/s"/>
                    <BSS Unit="bytes"/>
                    <m Unit="bytes"/>
                    <M Unit="bytes"/>
                    <EAR Unit="bit/s"/>
                    <PR1 Unit="bit/s"/>
                    <PR2 Unit="bit/s"/>
                </TrafficSpec>
            </TechnicalCharacteristic>
        </Option>
    </ServiceComponent>
</ApplicationProfile>
```

Figure 4-9: RealSystem XML profile with serviceID

4.2 Profile for NetMeeting

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Sotiris Maniatis, Haris
Tsetsekas (NTU) and John Karadimas (QSY) -->
<!DOCTYPE ApplicationProfile SYSTEM "ApplicationProfileV6.dtd">
<ApplicationProfile name="NetMeeting" version="3.01" type="MULTIMEDIA">
    <ServiceComponent type="AUDIO">
        <description>There are two Options for Audio</description>
        <NetworkCharacteristic>
            <description>H323 control port</description>
            <needProxy>No</needProxy>
            <lowerPortNo>1720</lowerPortNo>
            <isControlPort>Yes</isControlPort>
            <protocol>TCP</protocol>
        </NetworkCharacteristic>
        <NetworkCharacteristic>
            <description>The audio port is not fixed</description>
            <needProxy>Yes</needProxy>
            <nameProxy>NetMeeting</nameProxy>
            <protocol>RTP</protocol>
        </NetworkCharacteristic>
        <Option optionID="Audio_1" description="Audio half-duplex scenario">
            <SessionCharacteristic>
                <SessionCharacteristicParameter>
                    <name>Audio quality</name>
                    <description>perceived audio quality</description>
                    <qualifier>"normal"</qualifier>
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Connection</name>
                    <description/>
                    <qualifier>"half-duplex"</qualifier>
                </SessionCharacteristicParameter>
            </SessionCharacteristic>
            <TechnicalCharacteristic>
```

```
                <serviceID value="PCBR"/>
                <TrafficSpec>
                    <PR Unit="bit/s">60000</PR>
                    <BSP Unit="bytes">2000</BSP>
                    <SR Unit="bit/s"/>
                    <BSS Unit="bytes"/>
                    <m Unit="bytes">78</m>
                    <M Unit="bytes"/>
                    <EAR Unit="bit/s">20000</EAR>
                    <PR1 Unit="bit/s"/>
                    <PR2 Unit="bit/s"/>
                </TrafficSpec>
            </TechnicalCharacteristic>
        </Option>
        <Option optionID="Audio_2" description="Audio full-duplex scenario">
            <SessionCharacteristic>
                <SessionCharacteristicParameter>
                    <name>Audio quality</name>
                    <description>perceived audio quality</description>
                    <qualifier>"normal"</qualifier>
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Connection</name>
                    <description/>
                    <qualifier>"full-duplex"</qualifier>
                </SessionCharacteristicParameter>
            </SessionCharacteristic>
            <TechnicalCharacteristic>
                <serviceID value="PCBR"/>
                <TrafficSpec>
                    <PR Unit="bit/s">90000</PR>
                    <BSP Unit="bytes">2000</BSP>
                    <SR Unit="bit/s"/>
                    <BSS Unit="bytes"/>
                    <m Unit="bytes">78</m>
                    <M Unit="bytes"/>
                    <EAR Unit="bit/s">40000</EAR>
                    <PR1 Unit="bit/s"/>
                    <PR2 Unit="bit/s"/>
                </TrafficSpec>
            </TechnicalCharacteristic>
        </Option>
    </ServiceComponent>
    <ServiceComponent type="VIDEO">
        <description>There are three Options for Video</description>
        <NetworkCharacteristic>
            <description>H323 control port</description>
            <needProxy>No</needProxy>
            <lowerPortNo>1720</lowerPortNo>
            <isControlPort>Yes</isControlPort>
            <protocol>TCP</protocol>
        </NetworkCharacteristic>
        <NetworkCharacteristic>
            <description>The video port is not fixed</description>
            <needProxy>Yes</needProxy>
            <nameProxy>NetMeeting</nameProxy>
            <protocol>RTP</protocol>
        </NetworkCharacteristic>
        <Option optionID="Video_1" description="Video Low Quality scenario">
            <SessionCharacteristic>
                <SessionCharacteristicParameter>
                    <name>Video quality</name>
                    <description>perceived video quality</description>
                    <qualifier>"low"</qualifier>
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Window Size</name>
```

```
                <description>window size</description>
                <qualifier>"small"</qualifier>
            </SessionCharacteristicParameter>
            <SessionCharacteristicParameter>
                <name>Network Speed</name>
                <description>selected network speed</description>
                <qualifier>"fast"</qualifier>
            </SessionCharacteristicParameter>
        </SessionCharacteristic>
        <TechnicalCharacteristic>
            <serviceID value="PVBR"/>
            <TrafficSpec>
                <PR Unit="bit/s">160000</PR>
                <BSP Unit="bytes">2000</BSP>
                <SR Unit="bit/s">75000</SR>
                <BSS Unit="bytes">5120</BSS>
                <m Unit="bytes">60</m>
                <M Unit="bytes">1500</M>
                <EAR Unit="bit/s">75000</EAR>
                <PR1 Unit="bit/s"/>
                <PR2 Unit="bit/s"/>
            </TrafficSpec>
        </TechnicalCharacteristic>
    </Option>
    <Option optionID="Video_2" description="Video Medium Quality scenario">
        <SessionCharacteristic>
            <SessionCharacteristicParameter>
                <name>Video quality</name>
                <description>perceived video quality</description>
                <qualifier>"medium"</qualifier>
            </SessionCharacteristicParameter>
            <SessionCharacteristicParameter>
                <name>Window Size</name>
                <description>window size</description>
                <qualifier>"medium"</qualifier>
            </SessionCharacteristicParameter>
            <SessionCharacteristicParameter>
                <name>Network Speed</name>
                <description>selected network speed</description>
                <qualifier>"fast"</qualifier>
            </SessionCharacteristicParameter>
        </SessionCharacteristic>
        <TechnicalCharacteristic>
            <serviceID value="PVBR"/>
            <TrafficSpec>
                <PR Unit="bit/s">240000</PR>
                <BSP Unit="bytes">2000</BSP>
                <SR Unit="bit/s">130000</SR>
                <BSS Unit="bytes">6144</BSS>
                <m Unit="bytes">60</m>
                <M Unit="bytes">1500</M>
                <EAR Unit="bit/s">130000</EAR>
                <PR1 Unit="bit/s"/>
                <PR2 Unit="bit/s"/>
            </TrafficSpec>
        </TechnicalCharacteristic>
    </Option>
    <Option optionID="Video_3" description="Video High Quality scenario">
        <SessionCharacteristic>
            <SessionCharacteristicParameter>
                <name>Video quality</name>
                <description>perceived video quality</description>
                <qualifier>"high"</qualifier>
            </SessionCharacteristicParameter>
            <SessionCharacteristicParameter>
                <name>Window Size</name>
                <description>window size</description>
```

```
                    <qualifier>"large"</qualifier>
                </SessionCharacteristicParameter>
                <SessionCharacteristicParameter>
                    <name>Network Speed</name>
                    <description>selected network speed</description>
                    <qualifier>"fast"</qualifier>
                </SessionCharacteristicParameter>
            </SessionCharacteristic>
            <TechnicalCharacteristic>
                <serviceID value="PVBR"/>
                <TrafficSpec>
                    <PR Unit="bit/s">900000</PR>
                    <BSP Unit="bytes">2000</BSP>
                    <SR Unit="bit/s">700000</SR>
                    <BSS Unit="bytes">8192</BSS>
                    <m Unit="bytes">60</m>
                    <M Unit="bytes">1500</M>
                    <EAR Unit="bit/s">700000</EAR>
                    <PR1 Unit="bit/s"/>
                    <PR2 Unit="bit/s"/>
                </TrafficSpec>
            </TechnicalCharacteristic>
        </Option>
    </ServiceComponent>
</ApplicationProfile>
```

Figure 4-10: NetMeeting XML profile with serviceID

4.3 Profile for WinSip

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Anne Thomas (TUD) -->
<!DOCTYPE ApplicationProfile SYSTEM "ApplicationProfileV6.dtd">
<ApplicationProfile name="WinSIP" version="0.2.0" build="" type="VoIP">
    <ServiceComponent type="AUDIO">
        <description>Voice over IP application developped by Siemens Austria AG.
WinSip is a Windows application prototype which allows setup of full duplex
telephone calls in a point-to-point fashion. The calls are signaled using SIP. The
coded voice information is conveyed with RTP/UDP/IP. The available codec is PCM µ-
Law in this version.</description>
        <NetworkCharacteristic>
            <description>WinSip control port</description>
            <needProxy>No</needProxy>
            <lowerPortNo>5060</lowerPortNo>
            <isControlPort>Yes</isControlPort>
            <protocol>UDP</protocol>
        </NetworkCharacteristic>
        <NetworkCharacteristic>
            <description>We assume that the audio port is fixed</description>
            <needProxy>No</needProxy>
            <lowerPortNo>5004</lowerPortNo>
            <isControlPort>No</isControlPort>
            <protocol>RTP</protocol>
        </NetworkCharacteristic>
        <Option optionID="value" description="value">
            <SessionCharacteristic>
                <SessionCharacteristicParameter>
                    <name>Audio quality</name>
                    <description>perceived audio quality</description>
                    <qualifier>"normal"</qualifier>
                </SessionCharacteristicParameter>
            </SessionCharacteristic>
            <TechnicalCharacteristic>
                <description/>
                <serviceID value="PCBR"/>
```

```
                <QoSSpec>
                    <maxDelay Unit="ms">50</maxDelay>
                    <maxJitter Unit="ms"/>
                    <maxLoss Unit="float">0.1</maxLoss>
                    <bwGuarantee bwGuarantee="kb/s">64</bwGuarantee>
                    <ordering ordering="boolean">true</ordering>
                </QoSSpec>
                <TrafficSpec>
                    <PR Unit="bit/s">67500</PR>
                    <BSP Unit="bytes">2000</BSP>
                    <SR Unit="bit/s">64000</SR>
                    <BSS Unit="bytes"/>
                    <m Unit="bytes"/>
                    <M Unit="bytes"/>
                    <EAR Unit="bit/s">64000</EAR>
                    <PR1 Unit="bit/s"/>
                    <PR2 Unit="bit/s"/>
                </TrafficSpec>
            </TechnicalCharacteristic>
        </Option>
    </ServiceComponent>
</ApplicationProfile>
```

Figure 4-11: WinSip XML profile with both serviceID and QoSSpec

4.4 Profile for Unreal Tournament

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Anne Thomas (TUD) -->
<!DOCTYPE ApplicationProfile SYSTEM "ApplicationProfileV6.dtd">
<ApplicationProfile name="Unreal Tournament" version="current" build="" type="GAME">
    <ServiceComponent type="DATA">
        <description>Only data are exchange between the client and the server.
</description>
        <NetworkCharacteristic>
            <description/>
            <needProxy>No</needProxy>
            <nameProxy/>
            <lowerPortNo>7777</lowerPortNo>
            <upperPortNo/>
            <isControlPort>No</isControlPort>
            <protocol>UDP</protocol>
        </NetworkCharacteristic>
        <Option optionID="1" description="with LAN, cable moden, xDSL network
speed">
            <SessionCharacteristic>
                <SessionCharacteristicParameter>
                    <name>Gaming feeling</name>
                    <description/>
                    <qualifier>"good"</qualifier>
                </SessionCharacteristicParameter>
            </SessionCharacteristic>
            <TechnicalCharacteristic>
                <description/>
                <serviceID value="PMC"/>
                <QoSSpec>
                    <maxDelay Unit="ms">1000</maxDelay>
                    <maxJitter Unit="ms"/>
                    <maxLoss Unit="float">0.05</maxLoss>
                    <bwGuarantee bwGuarantee="kb/s">28,8</bwGuarantee>
                    <ordering ordering="boolean">true</ordering>
                </QoSSpec>
                <TrafficSpec>
                    <PR Unit="bit/s">20000</PR>
                    <BSP Unit="bytes">2000</BSP>
```

```
                <SR Unit="bit/s">10000</SR>
                <BSS Unit="bytes">10000</BSS>
                <m Unit="bytes">40</m>
                <M Unit="bytes"/>
                <EAR Unit="bit/s">10000</EAR>
                <PR1 Unit="bit/s"/>
                <PR2 Unit="bit/s"/>
            </TrafficSpec>
        </TechnicalCharacteristic>
    </Option>
    <Option optionID="2" description="with ISDN network speed">
        <SessionCharacteristic>
            <SessionCharacteristicParameter>
                <name>Gaming feeling</name>
                <description/>
                <qualifier>"medium"</qualifier>
            </SessionCharacteristicParameter>
        </SessionCharacteristic>
        <TechnicalCharacteristic>
            <description/>
            <serviceID value="PMC"/>
            <QoSSpec>
                <maxDelay Unit="ms">1000</maxDelay>
                <maxJitter Unit="ms"/>
                <maxLoss Unit="float">0.05</maxLoss>
                <bwGuarantee bwGuarantee="kb/s">28,8</bwGuarantee>
                <ordering ordering="boolean">true</ordering>
            </QoSSpec>
            <TrafficSpec>
                <PR Unit="bit/s">5000</PR>
                <BSP Unit="bytes">2000</BSP>
                <SR Unit="bit/s"/>
                <BSS Unit="bytes"/>
                <m Unit="bytes">40</m>
                <M Unit="bytes"/>
                <EAR Unit="bit/s"/>
                <PR1 Unit="bit/s"/>
                <PR2 Unit="bit/s"/>
            </TrafficSpec>
        </TechnicalCharacteristic>
    </Option>
    <Option optionID="3" description="with modem network speed">
        <SessionCharacteristic>
            <SessionCharacteristicParameter>
                <name>Gaming feeling</name>
                <description/>
                <qualifier>"poor"</qualifier>
            </SessionCharacteristicParameter>
        </SessionCharacteristic>
        <TechnicalCharacteristic>
            <description/>
            <serviceID value="PMC"/>
            <QoSSpec>
                <maxDelay Unit="ms">1000</maxDelay>
                <maxJitter Unit="ms"/>
                <maxLoss Unit="float">0.05</maxLoss>
                <bwGuarantee bwGuarantee="kb/s">28,8</bwGuarantee>
                <ordering ordering="boolean">true</ordering>
            </QoSSpec>
            <TrafficSpec>
                <PR Unit="bit/s">2600</PR>
                <BSP Unit="bytes">2000</BSP>
                <SR Unit="bit/s"/>
                <BSS Unit="bytes"/>
                <m Unit="bytes">40</m>
                <M Unit="bytes"/>
                <EAR Unit="bit/s"/>
```

```
                    <PR1 Unit="bit/s"/>
                    <PR2 Unit="bit/s"/>
                </TrafficSpec>
            </TechnicalCharacteristic>
        </Option>
    </ServiceComponent>
</ApplicationProfile>
```

Figure 4-12: Unreal Tournament XML profile with both serviceID and QoSSpec

4.5 Profile for Ultima Online

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Andreas Koenig (BAG) -->
<!DOCTYPE ApplicationProfile SYSTEM "ApplicationProfileV6.dtd">
<ApplicationProfile name="UltimaOnline" version="2" type="GAME">
    <ServiceComponent type="DATA">
        <description>Only data will be transferred between server and
client</description>
        <NetworkCharacteristic>
            <description/>
            <needProxy>No</needProxy>
            <nameProxy/>
            <lowerPortNo>2593</lowerPortNo>
            <upperPortNo/>
            <isControlPort>No</isControlPort>
            <protocol>TCP</protocol>
        </NetworkCharacteristic>
        <Option optionID="1" description="">
            <SessionCharacteristic>
                <SessionCharacteristicParameter>
                    <name>Gaming feeling</name>
                    <description/>
                    <qualifier>"good"</qualifier>
                </SessionCharacteristicParameter>
            </SessionCharacteristic>
            <TechnicalCharacteristic>
                <description/>
                <serviceID value="PMC"/>
                <QoSSpec>
                    <maxDelay Unit="ms">400</maxDelay>
                    <maxJitter Unit="ms"/>
                    <maxLoss Unit="float">0.02</maxLoss>
                    <bwGuarantee bwGuarantee="kb/s">64000</bwGuarantee>
                    <ordering ordering="boolean">1</ordering>
                </QoSSpec>
                <TrafficSpec>
                    <PR Unit="bit/s">2000</PR>
                    <BSP Unit="bytes"/>
                    <SR Unit="bit/s"/>
                    <BSS Unit="bytes"/>
                    <m Unit="bytes"/>
                    <M Unit="bytes"/>
                    <EAR Unit="bit/s"/>
                    <PR1 Unit="bit/s"/>
                    <PR2 Unit="bit/s"/>
                </TrafficSpec>
            </TechnicalCharacteristic>
        </Option>
    </ServiceComponent>
</ApplicationProfile>
```

Figure 4-13: Ultima Online XML profile with both serviceID and QoSSpec

5 Scenarios for the First Trial

The aim of the trial is to validate and verify the AQUILA network and mechanisms. This can be done at two levels: application level and network level. The following two points describe assessment methods and propose test scenarios at application level as well as network level for the first trial.

5.1 Application level QoS tests/validation scenarios

With the term "Application level QoS" we refer to the perceived quality parameters that are understandable by normal users. Examples of such parameters are the clearness of voice in IP telephony, or the quality of the viewed images in video streaming applications. As the following extract states, subjective and objective methods can be used, in order to assess the delivered quality of service:

Extract from [Watson96]:

*"**Assessing speech quality***

Speech quality can be measured either subjectively or objectively. Subjective methods (where 'subjective' refers to opinion rating and/or measurement of task performance) are agreed to be more reliable than objective methods involving instrumental assessment (...). The speech that is assessed in these tests is generally specific material, recorded or spoken under defined conditions.

Subjective speech assessments can be divided into two main groups, Mean Opinion Scores (MOS) and intelligibility tests.

Mean Opinion Scores (MOS) have traditionally been used in speech quality assessment in the telecommunications world where the speech that is assessed is at, or approaching, telephone or 'toll' quality speech. The MOS is typically a 5-point rating scale, covering the options Excellent, Good, Fair, Poor and Bad, and is the standard recommended by the CCITT (1984) (...). Listening subjects rate the quality of the speech they hear according to this scale. Other rating scales exist, for example scales which attempt to address the impairment of quality rather than the quality itself (...), but the MOS remains the most well-known and commonly used one.

Intelligibility tests have commonly been employed for the quality assessment of synthetic speech (...). These types of test usually entail the subject listening to parts of words (e.g. consonant-vowel tests), words (e.g. rhyme tests) or sentences (e.g. Harvard or Haskins sentences), and writing down what is heard, or answering comprehension questions on a passage of speech.

The application of either of these methods to assessing the quality of speech transmitted over the Internet has certain difficulties. These methods of assessing speech quality and intelligi-

bility have been developed for assessing the quality of telecommunications systems and synthetic speech. Their suitability for assessing speech quality over the MBone needs to be explored since packet loss of the sizes involved presents a novel type of degradation, the effect of which is hitherto unknown. In addition, the MBone is a highly unpredictable network in terms of quality of received information. Network load cannot be predicted beyond broad generalisations about peak loading times of day, and loss rates can alter drastically within the space of a few minutes or even seconds. Studies that have attempted to measure the loss characteristics of the Internet have found that packets tend to be lost individually rather than in clumps (…), but beyond this the Internet remains far more unpredictable than any other type of communication network. It is therefore difficult to apply traditional intelligibility tests to Internet speech, since there is no means of predicting whether or where the packet loss will occur. It is this constantly changing and unpredictable level of quality that presents complex problems for the evaluation of audio quality over the Internet."

It is not the aim of AQUILA to perform such assessment tests. They are very complex and require special equipment and knowledge. Therefore we suggest validating the QoS at application level by comparing qualitatively the speech quality produced by the applications by using:

- The best effort class
  - In unloaded network
  - In loaded network
- The appropriate network service (PCBR, PVBR, PMM, PMC)
  - In unloaded network
  - In loaded network

Each probing person should get the same questionnaire for every test and should not know during the test, which case he/she is dealing with. The questionnaire should be simple and basic.

The following chapters give some additional information on the concrete application scenarios.

5.1.1 RealSystem

Regarding the five typical different kinds of network connection, the scenarios are predefined. The streams have to be encoded with the original quality, and in dependence of the preferences for the SureStream encoding, the scenarios are given.

5.1.2 NetMeeting

Regarding the possible scenarios refer to the configuration possibilities in chapter 3.2.2.

5.1.3 WinSip

As WinSip offers only one operation mode, the number of possible scenarios are quite reduced: Two persons in two different rooms talk to each other via WinSip. This can be done with a headset or a loudspeaker and a microphone. To have the better results the two testing persons should speak in their mother tongue, and have the same one.

5.1.4 Unreal Tournament

For the trial scenario at least two *experienced* players with the same playing level should play the game over the AQUILA network. A dedicated server is not necessary. The players should not know by playing which QoS test case it is, and should answer for every test the same questionnaire.

5.1.5 Ultima Online

Ultima Online is a multi-player online game. At the Ultima Online World 200 up to 10000 users can play together. Such a world is typically installed on one up to ten servers. But this scenario is impossible to realise for the first trial. Therefore only one server and up to ten players are intended.

The testers need no experience in role playing games, cause they can increase the skills of their game characters by learning the game on "safe" places like cities. The testers can fight against monsters and other human players outside of the cities, when they get enough experience in playing Ultima Online on the safe places.

5.2 Network level QoS tests/validation scenarios

This series of tests involves quality parameters that are more tangible than the application level ones. Since network level QoS parameters are quantitative, they may be measured during the trials, by using appropriate equipment and procedures. The most important parameters to be measured are: throughput, delay, jitter (delay variability), and packet loss.

The general procedure for the validation of the AQUILA network concepts involves the following steps:

- First measure the appropriate QoS parameters by using the applications without QoS support. More specifically, the following scenarios can be envisaged[1]:
  - Influence the network in such a way that the throughput is less.
  - Influence the network in such a way that the probability for packet loss is high.
  - Influence the network in such a way that the probability for packet delay is high.

[1] The usefulness of the test cases depends on the QoS requirements of the concrete application to be tested.

- Influence the network in such a way that the probability for packet jitter is high.
- Mix the different influences.

In each scenario measure at application input the characteristics of the flow.

- Then use the applications (with exactly the same background load as before), but having requested QoS with the EAT.
- Compare the measured parameters.

5.2.1 RealSystem

RealSystem is the application system for streaming media. The factors that can severely impact the quality of streaming are: delay, jitter, bandwidth and packet loss. The probe parameter is packet loss (< 10%). Delay and jitter depend on the elected buffer size and the bandwidth. A higher delay is possible, because the RealPlayer is adjustable to buffer some data for reducing the problems with delay. The RealPlayer can automatically reduce the streaming level in case of problems with jitter.

5.2.2 NetMeeting

As mentioned before, it is better to disable (i.e. not to use) the data subsystem of NetMeeting, due to the intelligent bandwidth management features and the combination of streams' bandwidth it has incorporated.

NetMeeting has the same preconditions as WinSip, as far as audio is concerned. For the video part, three testing scenarios can be arranged: Low quality (small window, low-quality), Medium quality (medium window, medium quality), and Good quality (large window, best quality). (See also chapter 3.2.)

5.2.3 WinSip

WinSip is a typical VoIP application with only one service component: audio. Three factors can impact the quality of voice: Delay, jitter, and packet loss. Therefore, the tests/validation involving this application should take into account these characteristics of the application.

Acceptable parameters would be: packet loss (< 10%) and end-to-end delay (< 120 ms)

5.2.4 Unreal Tournament

As mentioned previously the game is like a lot of "Internet" applications programmed in such a way that it can cope with the actual best effort quality, by in this case predicting the game and executing the same code as the server. Only a few data transit between the server and the client takes place. However, low loss (< 5%) and low delay (< 100 ms) are required.

5.2.5 Ultima Online

Ultima Online is a game, which is programmed in order to cope with the actual best effort quality. A few data transit between the server and the clients is required with low packet-loss (< 2%) and low delay (< 150 ms).

6 Conclusions for the EAT and the second trial

The focus of the first trial is legacy applications, which cannot be modified and recompiled, respectively. These applications are not QoS-aware, i.e. they do not have any possibility to request for QoS. The support of such applications is therefore concentrated on manual QoS reservation requests via the GUIs of the EAT, and the Proxy that detects important information form the applications' data flow.

For a sufficient support of legacy applications it is necessary to analyse their real requirements belonging the network, its performance and its QoS. More specifically, it is necessary to get information about the technical characteristics (codecs, QoS requirements, bandwidth needs, etc.) and the network characteristics (used port numbers, etc.). Only by having such information, an adequate QoS support can be realised. In order to get this information and complete the reservation request *after* the application's start special proxies are needed, especially for dynamically negotiated network characteristics.

The basic support of legacy applications via the EAT middleware is therefore realised by the reservation GUIs and the Proxy. However, the application profile approach extends this concept by providing a mechanism that *keeps* the technical and some network characteristics for each application *persistent*. Furthermore, this mechanism allows reservation requests on a much more abstract level by preparing easy-to-understand QoS options (the so-called session characteristics) in order to present them to the "normal" end-user on the Legacy Application GUI. The application profile approach is flexible enough to cover a lot of different Internet applications with different service components (Audio, Video, Data) and different QoS requirements. This has been proved by the specification of five different profiles for the applications of the first trial. They contain all necessary information for reservation requests towards the RCL, and they can easily be modified and/or extended. Furthermore, new profiles can quickly be created on the base of the DTD and the provided examples.

The Legacy Application GUI – based on the profiles – is the main way for normal end-users to request for QoS. The EAT will include a Converter component which parses the chosen application profile and allows the dynamical construction of the appropriate GUI with the appropriate QoS options. The Converter forwards the technical characteristics towards the EAT Manager so that a complete reservation request can be prepared. The realisation of the GUI servlets and the Converter component are the next steps of the EAT's implementation.

However, the application profile approach may be re-used in a later phase. I might be useful not only for single legacy applications. While the first trial concentrates on basic Internet applications such as single streaming and conferencing tools, *complex* Internet services have to be considered for the second trial.

With the term "Complex Internet Service" a multimedia Web service is meant that consists of different "service components" within one Web site. For example, a complex Internet service can include a streaming service, a chat room, and so on. For such service components plug-ins, in fact legacy applications, are used. Application profiles can be used to provide suitable reservation scenarios.

The Web site/platform integrating such service components is then responsible to offer also the QoS options for its multimedia components. Moreover, this Web site is responsible to request for reservations at the EAT's API. This can be realised by different Web technologies such as Java servlets and Java Server Pages (JSPs). Thus, no modifications on the basic components (plug-ins) are needed.

In general, the EAT API is one way to offer QoS for the 2nd trial. Applications can request for QoS at **application level**. This will have several advantages:

- The application can directly interact with the EAT in order to get information regarding the offered QoS.
- The application can either adapt to this levels by requesting a predefined network service, or it can request for the custom service by specifying its QoS needs. (The EAT is then responsible for the mapping.)
- Several and different requests during the app's runtime are possible.
- A feedback is possible, concerning the reached QoS in contrast to the requested one.
- The end-user can directly be involved by presenting the QoS options for his/her choice.
- Application developers can be supported by providing a high-level QoS interface in order to create new, QoS-aware applications, or to modify existing ones.

Furthermore, the API itself or the ideas/architecture behind this API can be part of or based on standardisation activities, regarding the Java community, for example.

Another general way to provide QoS support for applications is at the **transport level**. This option is already used by the Proxy approach and can be extended by additional proxies depending on the application requirements. Examples are:

- RSVP Proxy
- SIP Proxy
- HTTP Proxy

For each of the protocols above a separate proxy is needed. They should be part of the existing Proxy framework.

7 Abbreviations

A

ACA	Admission Control Agent
API	Application Programming Interface
AQUILA	Adaptive Resource Control for QoS Using an IP-based Layered Architecture

B

BBG	Bertelsmann Broadband Group
BMG	Bertelsmann Music Group
BmS	Bertelsmann mediaSystems
BSP	Bucket Size for PR
BSS	Bucket Size for SR
BW	Bandwidth

C

CCITT	Consultative Committee for International Telegraph and Telephone
CIF	Common Intermediate Format
Codec	COmpression/DECompression

D

DSL	Digital Subscriber Line
DSP	Digital Signal Processor
DTD	Document Type Definition

E

EAR	Expected Average Rate
EAT	End-user Application Toolkit

F

FM	Frequency Modulation

G

GUI	Graphical User Interface

H

HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol

I

IETF	Internet Engineering Task Force
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
ITU	International Telecommunication Union

J

JPEG	Joint Photographic Experts Group
JSP	Java Server Pages

K

Kbps	Kilobits per second

M

M	Maximum Packet Size
m	minimum Policed Unit
Mbps	Megabits per second
MBone	Multicast Backbone
MOS	Mean Opinion Scores
MPEG	Moving Pictures Expert Group

N

NM	(Microsoft) NetMeeting
NS	Network Service

P

p2a	point-to-any
p2p	point-to-point
PBX	Private Branch eXchange
PCBR	Premium CBR
PCM	Pulse Code Modulation

PMC	Premium Mission Critical
PMM	Premium MultiMedia
PR	Peak Rate
PSTN	Public Switch(ed) Telephone Network
PVBR	Premium VBR

Q

QCIF	Quarter CIF
QoS	Quality of Service

R

RCL	Resource Control Layer
RFC	Request For Comments
RSVP	Resource Reservation Protocol
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
RTSP	Real Time Streaming Protocol
RTT	Round Trip Time

S

SIP	Session Initiation Protocol
SLA	Service Level Agreement
SMIL	Synchronised Multimedia Integration Language
SR	Sustainable Rate

T

TCL	Traffic Class
TCP	Transmission Control Protocol

U

UDP	User Datagram Protocol
UML	Unified Modelling Language
UT	Unreal Tournament

V

VCR	Video Cassette Recorder

VHS	Video Home System
VoIP	Voice over IP

X

XML	eXtensible Markup Language

8 References (all)

[Adanez] X. G. Adanez: Service Level Agreement from the User Perspective, Adventis Communications Engineering, Ch. Montillier 3a, 1009 Pully, Switzerland, http://www.adventis.ch/articles/sla_wp.htm

[Ambler98] S. W. Ambler. Building object applications that work, SIGS books, Cambridge university press, 1998

[D1202] IST-1999-10077-WP1.2-SAG-1201-PU-O/b0, System architecture and specification for first trial

[D2201] IST-1999-10077-WP2.2-TUD-2201-RE-O/b0, Specification of End-user Application Toolkit

[Eichler01] G. Eichler, F. Fünfstück, F. Ricciato, A. Thomas, C. Tsetsekas, M. Winter: Modern Software Engineering Methods for IP-QoS Resource Pool Management, Submitted paper for I²CS, January 2001

[Fowler98] M. Fowler: Analysis Patterns, Addison Wesley, 1998

[ITU2001] ITU World Telecommunication Policy Forum (WTPF), Report of the Secretary-General on IP Telephony, 2001, http://www.itu.int/wtpf/sgreport/finalreport31Jan.htm

[Mehrotra00] R. Mehrotra: Games People Play Online, *IEEE MultiMedia*, October-December 2000, p.18-20

[Nack01] F. Nack: Play the game, *IEEE MultiMedia*, January-March 2001, p.8-10, http://www.computer.org/multimedia/mu2001/pdf/u1008.pdf

[RFC2976] S. Donovan: The SIP INFO Method, http://www.ietf.org/rfc/rfc2976.txt

[VoIP] Voice over IP, White Paper, http://www.tothai.net/Thai/download/White%20Paper/voip.htm

[VoIP99] Voice over IP, Technical Paper, Fall 99, http://www.seas.smu.edu/~venkatra/Cse8344.doc

[Watson] A. Watson: Evaluating Real-Time Multimedia Audio and Video Quality, http://www.cs.ucl.ac.uk/staff/awatson/aw.htm

[Watson96] A. Watson, M. A. Sasse: Evaluating Audio and Video Quality in Low-Cost Multimedia Conferencing Systems, Interacting with Computers, 8 (3), pp. 255-275, 1996, http://www.cs.ucl.ac.uk/staff/awatson/iwc.html

[Watson97] A.Watson, M. A. Sasse: Multimedia Conferencing via Multicast: Determining the Quality of Service Required by the End User, Proceedings of AVSPN '97 – International Workshop on Audio-Visual Services over Packet Networks, 15-16 September 1997, Aberdeen, Scotland, pp. 189-194, 1997, http://www.cs.ucl.ac.uk/staff/awatson/avspn.html