

DESIGN AND EVALUATION OF A MULTI-USER VIRTUAL AUDIO CHAT

Maja Matijasevic

FER University of Zagreb, Unska 3, HR-10000 Zagreb, Croatia

E-mail: maja.matijasevic@fer.hr

Lea Skorin-Kapov

Ericsson Nikola Tesla, Krapinska 45, HR-10000 Zagreb, Croatia

E-mail: lea.skorin-kapov@etk.ericsson.se

Abstract – As advanced Internet services, networked virtual reality applications impose certain Quality of Service (QoS) requirements, due to rich multimedia content and perceived “real-time” interactivity. Two different representations of QoS are needed at the user/application level and at the communication level, which our approach attempts to relate using as a reference an interconnection model for networked virtual reality applications. We present the design and development of a multi-user virtual audio chat application and a performance evaluation based on QoS requirements. Objective and subjective methods for determining QoS have been addressed, and our results analysed for the purpose of possible QoS improvements.

Keywords: virtual reality, quality of service, subjective quality, multi-user application, audio chat

1. INTRODUCTION

In the area of advanced Internet services, networked virtual reality (NVR) may be considered a new teleservice based on merging multimedia computing and (tele)communication technologies. NVR applications developed for such purposes as entertainment, education/training, *e-commerce*, data visualisation and various simulations may be foreseen as having a tremendous market potential. Key characteristics of such applications are rich multimedia content and perceived “real-time” interactivity. Specifying these requirements in terms of QoS calls for different representations of QoS at the user/application level and at the communication level. While a user evaluates service performance in subjective and qualitative terms, communication level QoS is expressed in objective and quantifiable terms. In previous research, a method for relating high-level QoS characteristics with measurable application and communication parameters has been proposed [14, 13]. In this paper we present the design and development of a networked virtual reality service in the form of a multi-user virtual chat application. We address real-time audio streaming as one of the key communication capabilities used in virtual environments (VEs). QoS evaluation is performed in terms of both user level and communication level characteristics. The paper is organized as follows: Section 2 covers related work in which a reference model for NVR applications is proposed, along with a set of high level QoS characteristics for such applications. Section 3 covers the design and development of a multi-user virtual chat application. Section 4 describes a performance analysis of our developed application, including an evaluation of user level QoS characteristics and measurements of communication level parameters. Section 5 concludes the paper.

2. RELATED WORK

Modelling of networked VEs has been recognized as a difficult research problem. In general, research has mainly progressed in two complementary directions, one addressing the performance

of the virtual reality end-systems, the other addressing virtual reality communications.

The evaluation of the performance of VR systems has been recognized as a complex task due to a limited understanding of virtual interfaces and overall factors affecting performance. Previous research includes an in depth review of human factors evaluation techniques [11] where the authors present an overview of both objective and subjective performance measurements. A diagnostic tool called VRUSE was developed in the form of a ten-part questionnaire in which each part addresses a key usability factor of the interface (for example, functionality, user input, system output). A comprehensive overview of human factors issues in VEs can be found in [18]. User-based evaluation in VEs is addressed in [6], where a systematic study of the design, evaluation and application of VE interaction techniques is presented. Three basic task categories are defined (viewpoint motion control, selection and manipulation) and, for each, a number of interaction techniques are proposed. The authors identify a set of performance metrics addressing quantitative and qualitative factors for each task. A different approach to the evaluation of VEs, in which the focus in this case is on a particular application, is presented in [10] using usability engineering. The technique presented uses an iterative and structured user-centred design and evaluation approach to achieve a usable interface.

In terms of the communication requirements of NVEs, a comprehensive review of communication architectures, protocols and mechanisms may be found in [17]. Managing dynamic shared state and resource management have been identified as key issues for achieving good scalability and performance. In terms of networking QoS, bandwidth, latency, distribution schemes and reliability are considered critical [12]. Several methods developed for reducing bandwidth requirements, such as dead reckoning and area-of-interest based filtering, have been reviewed in [15]. In [7], the authors discuss a need for the development of a virtual reality transfer protocol, suitable for NVEs. An approach based on formal method, using extended fuzzy-timing Petri nets has been proposed in [21].

Limited research has been done on relating user perceived system performance with quantitative network parameters in distributed VEs. In [20], the authors discuss experiments in a collaborative VE project (COVEN) and observational evaluations of user-performance in network trials. There is, however, no formal modelling of the relationship between network and application level parameters. Our goal has been to address this area. When specifying requirements for a networked virtual reality environment, two complementary aspects of the system are taken into account. The application aspect is related to the question: how do the VE content and user interactions at the application level affect the communication characteristics? The communication aspect is related to the question: how are events at the communication level reflected at the application level? In order to relate the interactions at the application level and the traffic parameters needed for QoS, an interconnection model has been proposed as shown in Figure 1 [13].

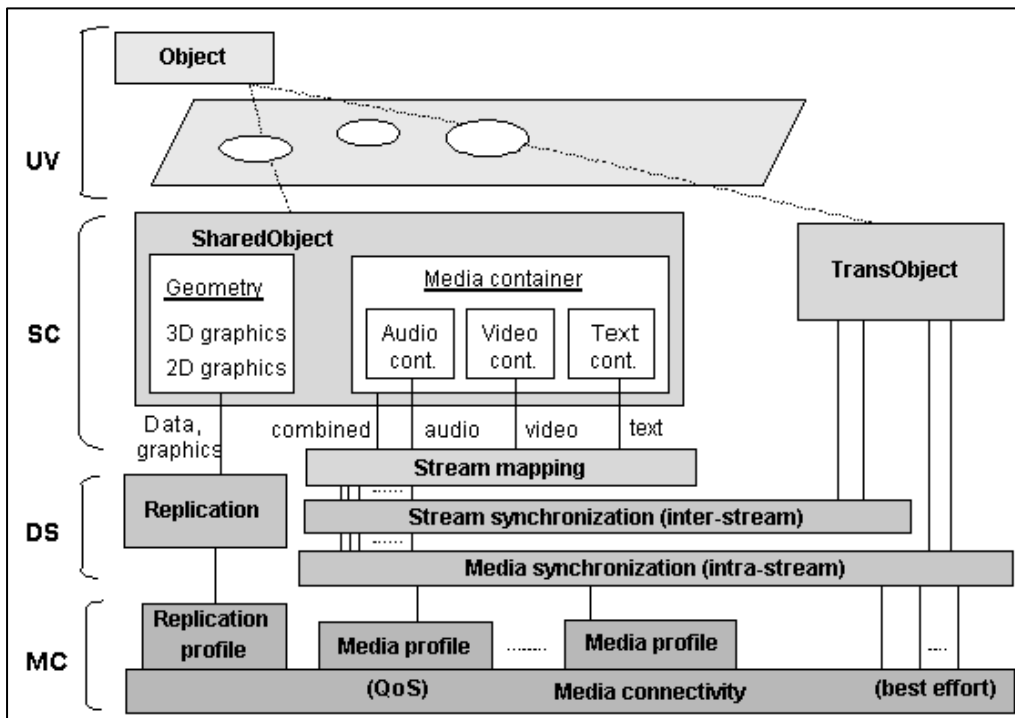


Figure 1. Interconnection model

The model shows mapping of virtual world objects from the user view (UV) to the media connectivity (MC) level. At the UV level, a VE is observed as a collection of virtual world objects, including the environment, static and dynamic objects and the embodiments (avatars) of other users or processes (autonomous agents). In this view, the distribution of the VE is transparent. The next two views represent different levels of composition and decomposition within the application: the spatial composition (SC), which focuses on spatial relationships; and the distribution and synchronization (DS) level, which focuses on temporal relationships between VE objects. The media connectivity (MC) level deals with mapping to transport level QoS parameters. Communication level QoS parameters related to NVR applications include band width (to support large volumes of media data and multiple users), latency and jitter (support for perceived "real-time" interactivity) and reliability. The notion of QoS at the user/application level is expressed differently from that at the communication level. In order to relate service performance as perceived by a user with quantitative, measurable application and communication parameters, a set of high-level QoS characteristics has been proposed [14]:

- **Interactivity:** Defined as the user's evaluation of the ability to interact with objects and other users in a distributed VE, as well as the extent of such interaction.
- **Immersion:** Defined [19] as the presentation quality of (multi)sensory information that a user perceives as a 3D synthetic environment. Thus, immersion may be quantified as a combination of visual, auditory and haptic immersion.
- **Plausibility:** Defined as the user's acceptance of events experienced in the virtual world as apparently reasonable and valid, based on one's experience.

Each of these three characteristics is quantified as a linear combination of quantities representing constituent characteristics. Interactivity is thus quantified as a linear combination of variables representing the sensory input support (tracking of user actions), navigation (user perception of movement in a VE) and user representation in the VE (including appearance and ability to interact with the environment). Immersion is quantified in terms of visual and auditory quality and their

synchronized presentation. Plausibility can be evaluated in terms of causality, or a cause-and-effect relationship (directly dependent on end-to-end delay) and consistency with the laws of physics (meaning, do the laws of physics, such as gravity, apply to our VE?). In this paper we will address some of these characteristics in further detail as applied to our developed application.

3. DESIGN AND DEVELOPMENT OF A MULTI-USER VIRTUAL CHAT

In order to address the needs of attractive NVR and multimedia services over the Internet and to better understand them in terms of QoS requirements, we developed a multi-user interactive prototype application. Virtual audio chat (VAC) enables real-time audio communication over the Internet between a multiple users using a combination of client/server and peer-to-peer (multicast) architecture. The user interface includes a Virtual Reality Modeling Language (VRML) model of a mobile telephone, allowing for the use of the interactive techniques implemented in the VRML.

3.1 VAC testbed configuration

The following hardware and software testbed has been used, as shown in Figure 2:

- WWW client – Pentium III (500 MHz, 128 MB RAM) running WinNT OS.
- Three WWW clients – Pentium III (800 MHz, 256 MB RAM) running WinNT OS.
- Two Sun Ultra5 workstations with 128 MB RAM running Solaris 2.6 OS; one serving as a WWW server and the other for performing measurements.

A 10 Mbit/s Ethernet LAN connects all hosts.

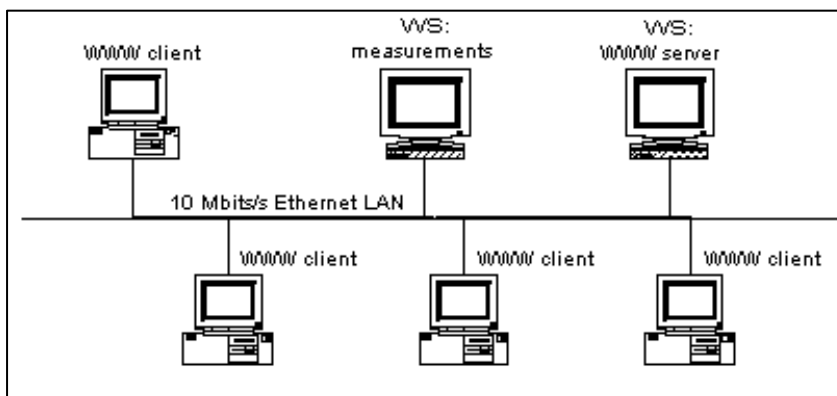


Figure 2. VAC testbed configuration

3.2 Development of VAC

Our VAC application consists of three basic components:

- A (modified) Session Directory (*sdr*) tool [5] which enables a user to schedule and announce a multimedia session on the Multicast Backbone (MBone);
- A VRML model of a mobile phone;
- A Java applet that opens a Real-time Transport Protocol (RTP) [16] based audio conference.

A user wishing to initiate a session starts *sdr* and defines all of the necessary session parameters, such as the time the session is active and the media comprising the session. Once the parameters have been defined and the session announced, users connected to the Internet and running the *sdr*

tool can choose to join (unless permission to join is denied), thus launching a Web browser from sdr and downloading a Web page from the server. The page contains a VRML model of a mobile phone. By way of user interactions implemented in VRML, the user can enter into the phone the multicast address and port number specified by sdr. A Java applet then opens an RTP-based audio conference using this multicast address and port number.

3.2.1 VRML model of a mobile phone

VRML is a file format that provides the means for describing interactive 3D worlds with integrated multimedia content in a distributed environment such as the Internet [1]. The VRML is capable of representing static and animated dynamic 3D and multimedia objects with hyperlinks to other media elements. Figure 3 shows a snapshot of our developed VRML model based on the Ericsson T10 mobile phone.

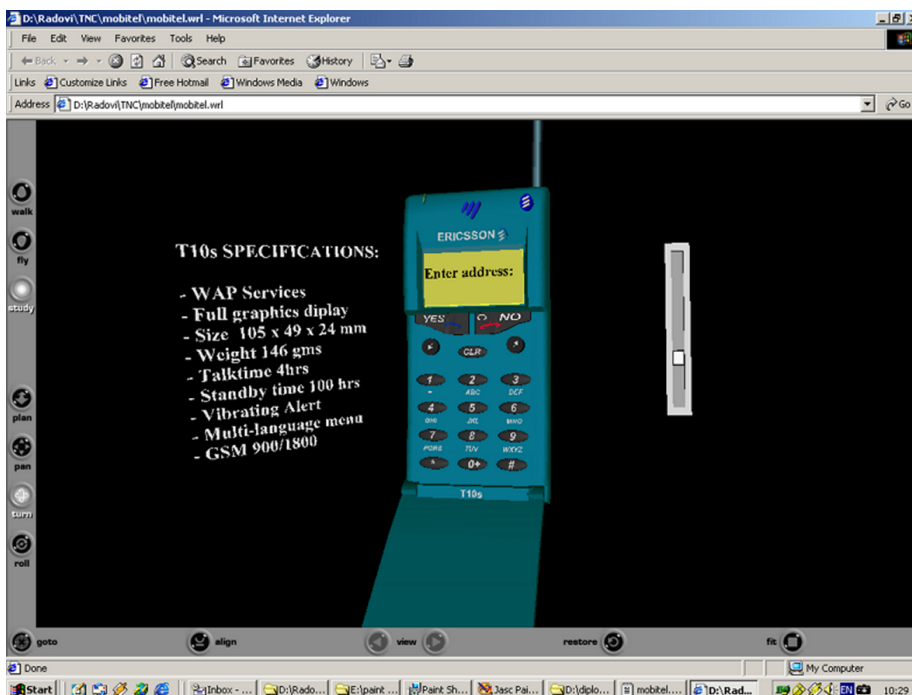


Figure 3. VRML model of mobile phone

User interactions include the following:

- Opening and closing of the active flip.
- Interaction with the virtual keys.
- Lighting of the display.
- User control of buffer size during audio streaming.

Interactions and dynamic animations were enabled through the use of various Sensor nodes and Script nodes implemented in the VRML. Much use was made of the touch sensor, used to generate events in response to the position of the mouse or to mouse clicks on certain objects. Scripts written in JavaScript and Java were used to define how events related to user interactions were to be processed. An interesting part of our VE is a scroll bar representing a virtual media-data buffer. By setting the position of the bar, the user is actually controlling the amount of audio data that is buffered prior to being passed on to the next processing stage. This can be used to reduce the effect of jitter, or variations in latency, as perceived by the user. In our LAN measurements, however, buffering did not produce any perceivable effects.

3.2.2 Real-time audio communication through the Internet

In order to join or initialise an audio chat on the Internet, a user enters a multicast address and port number by pressing the numbers on the virtual mobile phone with a mouse. The integration of the VRML and Java allows us to make use of the Java Media Framework (JMF) API that provides a way for audio, video and other time-based media to be added to Java applications and applets [3]. An application for transmitting and receiving RTP data has been programmed to enable an audio conference between multiple users. The new RTPManager Java class defines methods for initialising, running, and closing an RTP session.

The RTP [16] provides end-to-end delivery services for data with real-time characteristics such as interactive audio and video. It is designed so that the application may control the loss detection and recovery of packets (usually handled by lower levels in the communication architecture), enabling the use of simpler, unreliable protocols at the network and transport level. RTP works in combination with the RTP Control Protocol (RTCP) that monitors data delivery and provides control over data transport. RTP does not address resource reservation and does not guarantee QoS for real-time services.

3.2.3 Modified session directory tool

Session Directory (sdr) is a tool designed for scheduling and announcing multimedia sessions on the MBone, a virtual multicast network that uses the physical infrastructure of the Internet and implements virtual multicast in software. A session may consist of a number of multimedia components, such as audio, video and shared text editing. Each component is represented by an appropriate user application that may be launched from sdr. The sdr implements the Session Description Protocol (SDP) [8] for the purpose of describing a multimedia session, and the Session Announcement Protocol (SAP) [9] for the announcement of a session. In order to add support for our VAC application, we needed to modify sdr by enabling a new type of media, vrml_audio as seen in Figure 4.

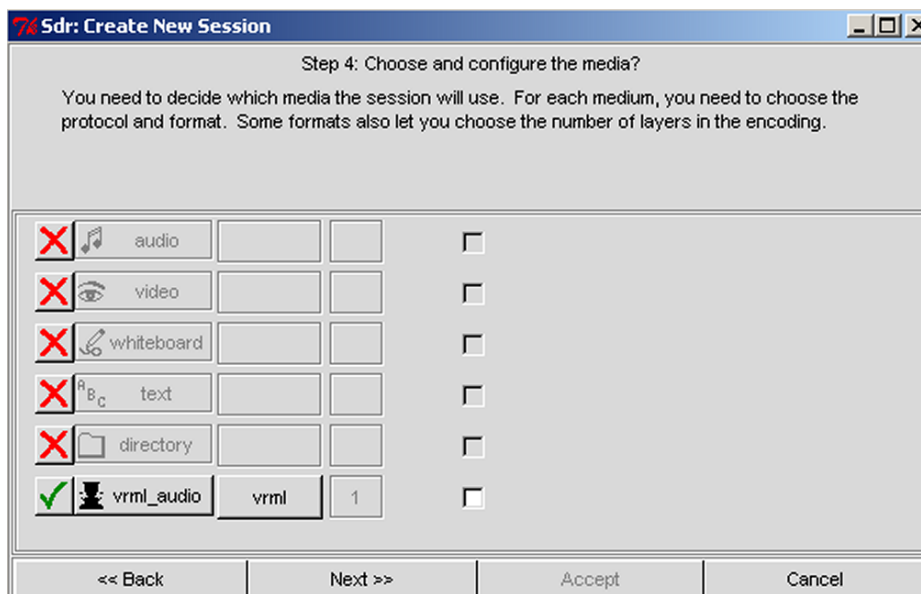


Figure 4. Choosing media components when initiating a session

A new plug-in configuration file was written, associating a new media type vrml_audio, protocol RTP, user application vrml_audio.exe and file format vrml. When a user joins a session containing the media component vrml_audio, sdr then knows which user application to run. In this case, a Web browser is launched and a Web page containing our virtual mobile phone is downloaded.

3.2.4 Problems encountered and solved

Challenges encountered during the development phase of the application components included adding support for user interactions and modification of the sdr tool. After initial development, the problem was to achieve interaction between these components. When a user ran sdr and joined a "vrml-audio" session, sdr had to launch vrml_audio.exe. This information needed to be included in the plug-in configuration file mentioned above when modifying sdr. Secondly, it proved difficult to directly translate the session multicast address and the port number specified by sdr to the virtual phone model. The users were told to manually enter the address and port by clicking the virtual keys. This problem may be solved by passing the address and port as parameters from sdr to vrml_audio.exe. From vrml_audio.exe, the virtual world can be downloaded and the RTP streaming program run with the received parameters. Providing multicast support for our RTP-based audio chat caused no real problems due to the JMF API support for multicast distribution.

3.2.5 The Interconnection Model

The previously described interconnection model for NVR applications as applied to our VAC application is shown in Figure 5.

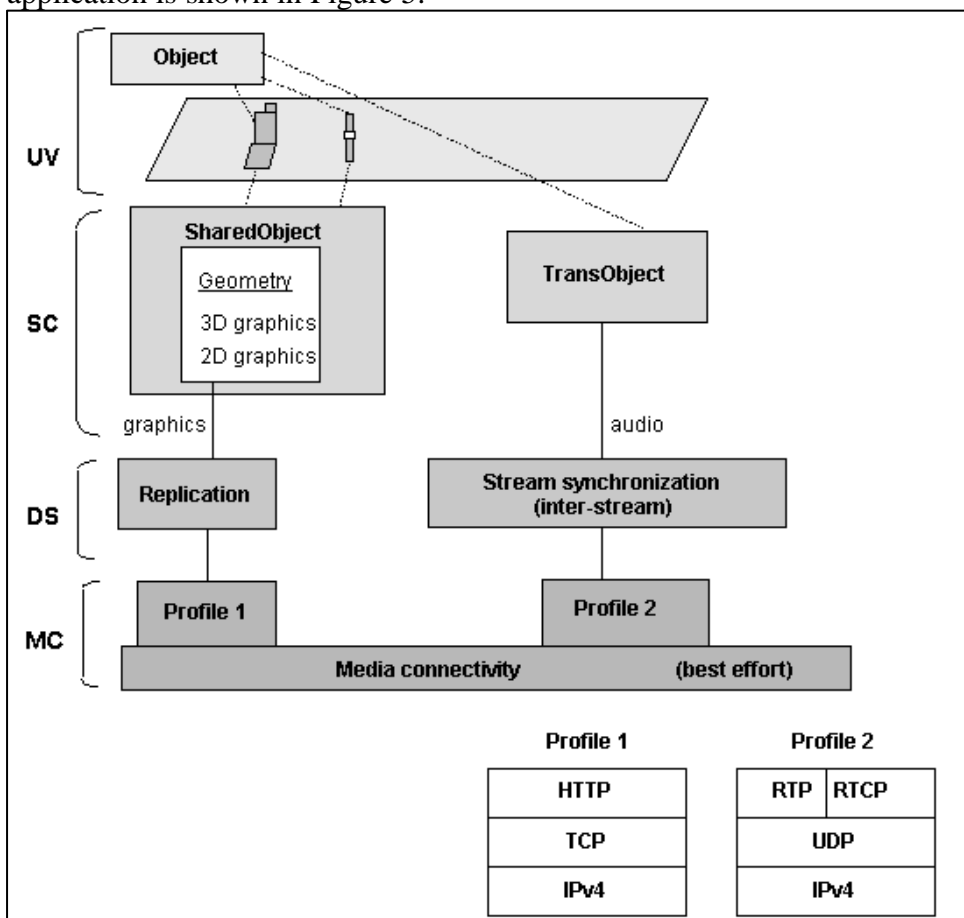


Figure 5. VAC interconnection model

The user view corresponds to the user's experience of the VE. Objects in the VE can be classified as SharedObjects and TransObjects. SharedObjects require replication and contain geometry, along with a media container that defines spatial attributes for each data type. TransObjects contain continuous media and do not require replication, establishing instead a streaming connection between the source and the sink. In our case, the mobile phone and the virtual media-data buffer can be considered SharedObjects, while the TransObject contains the audio stream without any spatial attributes. Replication is only necessary when the VE is initially downloaded, allowing each user to further manipulate his/her own local environment. Inter-stream synchronization refers to synchronization within the audio stream and is implemented in the RTP protocol. Profiles 1 and 2 represent a combination of the protocols used. The QoS at the communication level is "best-effort" due to the fact that resource reservation protocols were not implemented in our laboratory network.

4. PERFORMANCE ANALYSIS

Measurements of communication level parameters and an evaluation of user-level QoS characteristics were performed in the previously described laboratory network environment. A group of nine representative users were asked to run and evaluate our VAC application.

4.1 RTP and RTCP packet throughput

In order to measure RTP/UDP and RTCP/UDP packet throughput, we used the Ethereal program (version 0.8.16) [2]. Ethereal enabled us to capture, filter and analyse network traffic for the duration of our audio chat. Four users simultaneously took part on four different PCs in our laboratory network. Measurements were conducted according to the following steps:

- Run sdr.exe on four client PCs.
- Announce a VAC using sdr on one client (remaining three users receive notification).
- Start Ethereal capture on workstation designated for measurements.
- Users join announced session through sdr interface.
- Users enter multicast address (designated by sdr) and begin RTP streaming.
- Audio chat between four users.
- End of chat and closing of conference.
- Stop Ethereal capture.

In Figure 6 we see an example of throughput measured on one participating client. The entire packet size was taken into account, including all headers (Ethernet, IP, UDP, RTP).

Figure 6. RTP and RTCP packet throughput

The beginning of RTP packet generation indicates the moment when the user joined the audio chat (after entering the multicast address onto the mobile phone). In this particular case, the gap we see indicates that the user left the chat and then rejoined after a certain period of time. The oscillation in the RTP traffic curve is related to a chosen time interval of 1 second for calculating throughput and to the effects of delay jitter (the choice of a finer time interval would have resulted in a "straighter" curve). Parallel to the generation of RTP traffic, we see the occasional generation of RTCP packets with information regarding QoS. Results such as these relating to measured throughput and RTCP packet information can be used in the future for the purposes of resource reservation.

4.2 Interactivity

In evaluating user-level QoS, we have addressed interactivity as relating to the scope and extent of a user's interactions in a VE. An empirical approach using a questionnaire was used to obtain user evaluation. The group of users taking part in VAC was asked to evaluate on a scale of 1 to 10 the functionality and interaction techniques relating to our VE. An average score of 6.11 was obtained, with most users commenting on the difficulty of entering the address onto the phone due to trouble with positioning the mouse pointer. A possible improvement would be to enable entering the address using a keyboard. When asked to evaluate the ability to navigate in the VE, on a scale of 1 to 10, an average score of 9.00 was determined. Navigation techniques are implemented in the VRML browser, allowing us to conclude that the users were satisfied with the interface.

Interactivity can be quantified as a combination of user evaluations for sensory support, navigation, and user representation. Quantitative parameters and their values relating to our VAC application are shown in Table 1.

Parameter	Value
Tracking	Mouse tracking Virtual controls
Command input	Pointer (2D) recognition
Navigation technique	Examine, fly, walk
User appearance (avatar quality)	None (viewer)
Interactions	Object selection and manipulation

Table 1. Quantitative interactivity parameters

The keys on our mobile phone can be considered virtual controls. The tracking of user interactions refers to 2D-pointer mouse tracking.

Due to the fact that the user interface is the part of the application responsible for interactivity, a parameter related to user interface performance is initialisation time. Initialisation time corresponds to the time necessary to download the initial VE from a Web server. Measurements have shown that a significant part of this time is due to the initialisation of the VRML plug-in.

4.3 Immersion

Immersion has been defined in terms of the visual and auditory quality of a VE and their synchronized presentation. Users were asked to evaluate the visual quality of our VRML mobile phone model on a scale of 1 to 10, resulting in an average score of 8.89. All users agreed that the model shows great resemblance to the original Ericsson T10 mobile phone. They were also asked to comment on perceived auditory quality in which case most users described the sound as "clear and recognizable speech, but with slight delay".

Quantitative parameters related to immersion and their values relating to our VAC application are

shown in Table 2.

Parameter	Value
Sound quality	Phone quality (8000 Hz, 8 bit/sample)
Sound spatialization	No

Table 2. Quantitative immersion parameters

In our case, there is no spatial component related to sound. A possible improvement would be to associate sound with a Sound node implemented in the VRML, relating the volume of the sound with the position of the mobile phone in the virtual world.

Latency is a network parameter that influences user perception of sound quality, expressed in terms of total latency along the communication path and latency variance (jitter). The nature of real-time applications is such that packet loss is often tolerated better than delay, requiring a trade-off between latency and reliability. We performed measurements of jitter and packet loss while running our VAC application using the rtpmon tool (version 1.0a7) [4]. Rtpmon reads RTCP packets generated by users taking part in a specified session. RTCP packets sent out by each participant provide feedback to other participants about reception quality. The tool sorts out, filters and displays statistics related to jitter and packet loss. By using this tool, it is possible to monitor a session, and recognize and diagnose problems related to each individual receiver. By running rtpmon parallel to four users taking part in VAC, the main rtpmon window is opened as shown in Figure 7.

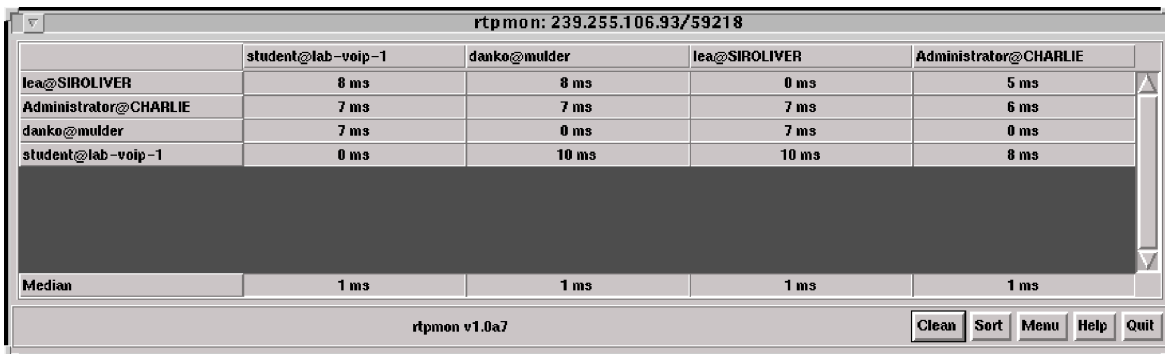


Figure 7. Main rtpmon window

We see here a table of receivers (rows) and transmitters (columns) and the most recent values received by rtpmon relating to jitter. According to this data, values ranged from 0-10 ms. Other windows showed statistics related to packet loss, of which there was none in our LAN environment. Audio data has been buffered in order to reduce the effects of jitter. Because of the relatively small values measured for jitter, the effects of changing buffer size (by way of virtual scroll bar) remained unnoticed. By running VAC between users on different subnets, jitter due to additional processing and routing would have greater effect.

4.4 Plausibility

The cause-and-effect relationship is directly related to the distribution of users and end-to-end delay at the communication level. In our case, all users were located in a LAN environment and total latency was measured using the ping program. Measurements showed round-trip-time (RTT)

as being ≤ 1 ms. Perceived latency during our audio chat can therefore be associated with application level processing, rather than with communication level delay.

5. CONCLUSION

In this paper we have addressed NVR services over the Internet and attempted to better understand them in terms of QoS requirements. For this purpose we presented the design and implementation of a multi-user interactive prototype application, virtual audio chat (VAC), and evaluated performance in terms of both user level QoS characteristics and communication level QoS parameters. Performance results may be used for the purposes of application improvements (as perceived by the user) as well as for resource reservation. The overall approach, from model to implementation, is general and may be applied to a wide range of virtual reality applications.

6. ACKNOWLEDGEMENTS

This research was partially funded by Ericsson Nikola Tesla Company, Zagreb, Croatia, under research project R00101 "Networked Virtual Reality". The authors would also like to thank anonymous reviewers for their comments and suggestions on how to improve the paper, and Ms. Carol de Groot of TERENA for editing the final text version of the paper.

7. REFERENCES

- [1] —, Information Technology – Computer graphics and image processing – The Virtual Reality Modeling Language (VRML) – Part 1: Functional specification and UTF-8 encoding, ISO/IEC 14772-1:1997, 1997
- [2] --, Ethereal, free network protocol analyzer,
Available from: <http://www.ethereal.com>
- [3] --, Java Media Framework API,
On-line documentation: <http://www.java.sun.com/products/java-media/jmf>
- [4] --, Rtpmon: a Third-Party RTCP Monitor,
Available from: <http://bmrc.berkeley.edu/~drbacher/projects/mm96-demo/index.html>
- [5] --, Session Directory Tool, in Mbone Conferencing Applications,
Available from: <http://www-mice.cs.ucl.ac.uk/multimedia/software/sdr>
- [6] D. A. Bowman, L. F. Hodges, Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments, *Journal of Visual Languages and Computing* 10:1 (1999) 37-53
- [7] D. Brutzman, M. Zyda, K. Watsen, and M. Macedonia, Virtual reality transfer protocol (vrtp) design rationale, in: *Proc. WET ICE: Sharing a Distributed Virtual Reality*, Cambridge, MA, June 1997, pp. 179-186
- [8] M. Handley, V. Jacobson, SDP: Session Description Protocol, Internet RFC 2327, April 1998
- [9] M. Handley, C. Perkins, E. Whelan, SAP: Session Announcement Protocol, Internet RFC 2974, October 2000
- [10] D. Hix, J. E. Swan II, J. L. Gabbard, M. McGee, J. Durbin, T. King, User-Centered Design and Evaluation of a Real-Time Battlefield Visualization Virtual Environment, in: *Proc. of IEEE Virtual Reality '99*, 1999, pp. 96-103
- [11] R. S. Kalawsky, S. T. Bee, S. P. Nee, Human Factors Evaluation Techniques to Aid Understanding of Virtual Interfaces, *BT Technology Journal* 17:1 (1999) 128-141
- [12] M. R. Macedonia, M.J. Zyda, A taxonomy for networked virtual environments, *IEEE Multimedia* 4:1 (1997) 48-56
- [13] M. Matijasevic, D. Gracanin, K. P. Valavanis, I. Lovrek, A Framework for Multi-user

- Distributed Virtual Environments, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 32:4 (2002) 416-429
- [14] M. Matijasevic, D. Gracanin, K. Valavanis, I. Lovrek, Towards QoS specification for distributed virtual environments, in: *Proc. of the 10th Mediterranean Electrotechnical Conference Melecon 2000, Vol. I, 2000*, pp.132–135
- [15] B. Roehle, Channeling the data flood, *IEEE Spectrum* 34:3 (1997) 32–38
- [16] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, *Internet RFC 1889*, January 1996
- [17] S. Singhal, M. Zyda, *Networked Virtual Environments: Design and Implementation*, ACM Press Siggraph Series and Addison-Wesley, New York, NY, 1999
- [18] K. M. Stanney, R. R. Mourant, R. S. Kennedy, Human Factors Issues in Virtual Environments: A Review of the Literature, *Presence* 7:4 (1998) 327–351
- [19] R. Stuart, *The Design of Virtual Environments*, McGraw/Hill, Inc., New York, 1996
- [20] J. Tromp, A. Bullock, A. Steed, A. Sadagic, M. Slater, E. Frecon, Small Group Behaviour in the COVEN Project, *IEEE Computer Graphics and Applications* 18:6 (1998) 53-63
- [21] Y. Zhou, T. Murata, T. A. DeFanti, Modeling and Performance Analysis Using Extended Fuzzy-Timing Petri Nets for Networked Virtual Environments, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 30:5 (2000) 737-756

7. VITAE

Maja Matijasevic received her Dipl.-Ing. (1990), M.Sc. (1994), and Ph.D. (1998) degrees in Electrical Engineering from the University of Zagreb, Croatia, and the M.Sc. in Computer Engineering (1997) from the University of Louisiana at Lafayette, LA, USA. She is presently an Assistant Professor in the Department of Telecommunications, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia, which she initially joined in 1991. She has been a visiting researcher at the University of Louisiana at Lafayette, LA, USA, and at the Technical University of Graz, Austria. Her main research interests include networked virtual environments and advanced multimedia services for next generation networks. Ms. Matijasevic is a member of IEEE, ACM, and Upsilon Pi Epsilon Honor Society in the Computing Sciences.



Lea Skorin-Kapov received her Dipl.-Ing. (2001) degree in Electrical Engineering from the University of Zagreb, Croatia. She is employed as a research engineer in the Research and Development Center of Ericsson Nikola Tesla in Croatia, and is currently working towards her Masters degree in the area of networked virtual reality. Her main research interests include quality of service, streaming and virtual reality.