End-to-end QoS for Virtual Reality Services in UMTS

Lea Skorin-Kapov
R & D Center
Ericsson Nikola Tesla
Zagreb, Croatia
lea.skorin-kapov@etk.ericsson.se

Dario Mikić
Department of Telecommunications
FER, University of Zagreb
Zagreb, Croatia
dario.mikic@fer.hr

Darko Huljenić
R & D Center
Ericsson Nikola Tesla
Zagreb, Croatia
darko.huljenic@etk.ericsson.se

***Abstract*— Virtual reality (VR) services may be considered a good representative of advanced services in the new generation network. The focus of this paper is to address Quality of Service (QoS) support for VR services in the context of the UMTS QoS framework specified by the 3G standardization forum, the Third Generation Partnership Project (3GPP). We propose a classification of VR services based on delivery requirements (real time or non-real time) and degree of interactivity that maps to existing UMTS QoS classes and service attributes. The mapping is based on matching VR service requirements to performance parameters and target values defined for UMTS applications. Test cases involving heterogeneous VR applications are defined using as a reference a general model for VR service design and delivery. Measurements of network parameters serve to determine the end-to-end QoS requirements of the considered applications, which are in turn mapped to proposed VR service classes.**

Keywords-Virtual Reality Service, Quality of Service, UMTS, Service Classification

I. INTRODUCTION

The Next Generation Network (NGN) is expected to support a great variety of applications with different quality of service (QoS) requirements. Virtual Reality (VR) services, representing a step ahead of "traditional" multimedia, aim to provide an advanced human computer interface allowing users to interact with a simulated 3D environment. Such services are characterized by high quality 3D graphics, integrated multimedia components, support for multiple users, and unpredictable traffic flows due to dynamic user interactions. Great market potential for VR may be foreseen in areas such as entertainment, e-commerce, education/training, various simulations, and data visualizations.

Standards proposed by the Third Generation Partnership Project (3GPP) define a layered QoS architecture to be used in the Universal Mobile Telecommunications System (UMTS) [2]. Specifications define four UMTS QoS classes intended to cover a broad range of applications primarily distinguished based on delay sensitivity [1]. The UMTS QoS classes are defined at the application layer in terms of end-to-end performance expectations, and are supported through interaction of underlying bearer services. Within existing specifications, there is no clear description provided of the requirements of VR services. Our purpose in this paper has been to propose a classification of VR services based on delivery requirements (real time or non-real time) and degree of interactivity that maps to existing UMTS QoS classes. The mapping is based on matching VR service requirements to performance parameters and target values defined for UMTS applications.

Case studies have been conducted in an emulated network environment to determine the end-to-end requirements of a number of heterogeneous VR applications, which serve to verify the proposed service classification and mapping. The applications used as test cases are considered in the context of a general reference model proposed for VR service design and delivery [12]. The model defines key service parameters in the form of "service profiles" needed for VR service adaptation in response to end user capabilities/preferences. Test case applications are classified as belonging to VR service classes based on measurements of network parameters defined in the service profile. Where possible, adaptation of service parameters enabled us to consider the implementation of different service versions with variations in application/network level QoS requirements.

The paper is organized as follows. A discussion of related literature and standards, focusing on VR service requirements and UMTS QoS classes is given in Section II. A mapping of VR services to UMTS QoS classes is proposed in Section III. Section IV covers test cases providing experimental verification of the proposed service classification. Section V gives a discussion of achieved results. Section VI concludes the paper.

II. OVERVIEW OF RELATED LITERATURE AND STANDARDS

In this section we briefly discuss two areas: issues relating to VR service requirements, and QoS classes for UMTS as defined in standards.

A. *VR service requirements*

VR applications present a challenge for multimedia networking for a number of reasons. These include providing support for multiple distributed users (sometimes up to tens or hundreds) taking part in a shared virtual environment using dynamic forms of communication, and the integration of media components such as 3D graphics, text, streaming audio and video [7]. Key network issues relating to NVR applications have been identified as bandwidth, network distribution, latency, and reliability [9].

Available bandwidth is one of the fundamental issues dictating the possible size and richness of a virtual environment. The required bandwidth corresponding to a VR service depends on the number of users, the distribution architecture (unicast, multicast, broadcast), and the VR/multimedia content. The network distribution scheme used affects the scalability of a virtual environment (VE), with multicast distribution providing a scalable solution. Another important issue is latency, expressed in terms of delay and delay variance (jitter). Latency requirements stem from user perception of "real-time" interactivity with a VE and other users or autonomous processes in the environment. Reliability is an issue that often forces a compromise between bandwidth and latency, with real time applications often tolerating loss better than delay.

As an important subset of VR services, Collaborative VEs (CVEs) impose certain requirements due to the fact that collaboration is often a real time and highly reactive multi-user process [15]. Virtual Prototyping System (VPS) is an example in the area of collaborative engineering with an estimated maximum delay requirement of 100 ms for collaborative virtual prototyping [8]. A different study suggested that 200 ms delay might be considered acceptable in a CVE, as long as there is practically no jitter [13]. The authors showed that in a session involving object manipulation tasks, performance in a network with 10 ms delay and considerable jitter was nearly the same as in a network with 200 ms delay and no jitter. Generally quoted requirements for distributed simulations with tightly coupled interactions are 100 ms delay for update messages and 50 ms jitter [14].

Requirements depend on the type and purpose of the application. CVEs designed for purposes such as performing complex manipulative tasks require minimal latency and high bandwidth, while many CVE applications are designed so that users spend most of the time navigating in 3D space and less time manipulating objects and interacting with others. Such CVEs may be considered more tolerant with regards to network latency [13].

Numerous multiplayer computer games (e.g. Unreal, Quake) take place in a VE and are therefore considered networked VR applications. Examples include sports games, first person shooter (FPS) games, and real time strategy (RTS) games. Bandwidth requirements depend on the number and distribution of users, along with the applied distribution scheme. In an RTS game, latency up to 500 ms may be considered acceptable (as long as jitter is low), while games requiring tight hand-eye motor control such as FPS demand that latency remains less than 100 ms. It has been argued that a player's success in an FPS game can be critically affected with 'ping times' (between client and server) measuring over 150 ms [17].

The integration of streaming audio/voice or video into a VE may enhance the sense of immersion perceived by users in applications such as virtual conferencing, CVEs, or multiplayer games. Overall quality is dependent on both quality of auditory and visual information, as well as synchronization quality. Important for VEs is the notion of animation streaming, in which case information regarding facial or body positions of avatars is streamed over the network. Delay, jitter, and synchronization with other media components are factors influencing overall quality.

B. *Existing QoS standards: UMTS QoS classes*

ITU-T recommendation F.700 [4] provides a general methodology for constructing multimedia services where a service is decomposed into a set of communication tasks, each of which handles a set of media components. In [5], a model for multimedia QoS categories is defined taking into account user expectations for a range of multimedia applications (Figure 1).

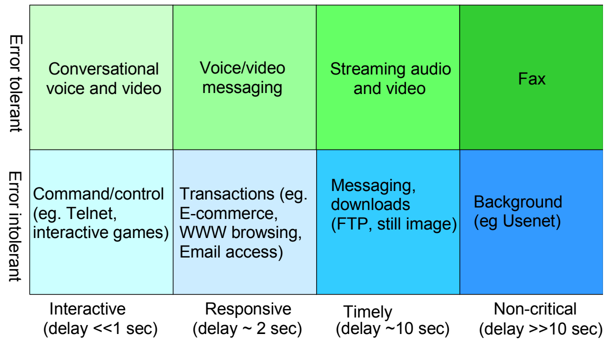

Figure 1. Model for user-centric QoS categories

In accordance with this model, 3GPP specifications [2] have classified UMTS services into four different QoS classes depending on the delay sensitivity of application traffic. A brief description of each class is given.

- *Conversational class*: represents conversational streaming applications that are very delay sensitive. Examples include telephony speech, voice over IP, and video conferencing. Limits for acceptable transfer delay are very strict, along with requirements on preserving the time relation between different stream entities.
- *Streaming class*: represents real time streaming applications that are primarily unidirectional. This scheme applies when the user is looking at (listening to) real time video (audio). The class is characterized by limited delay variations for end-to-end flows, with no requirements on low transfer delay.
- *Interactive class*: represents the classical data communication scheme characterized by the request-response pattern of the end user. Example applications include Web browsing, data base retrieval, and server access. A key characteristic for QoS is low bit error rate for transferred packets.
- *Background class*: the fundamental characteristic of this class is that the destination is not expecting the data within a certain time. Data can be sent and received in the background, with low bit error rate and no specific requirements on delay.

The defined classes are specified at the application level with an outline given of end user/application QoS requirements for example applications in terms of key performance parameters and target values. Indicated are the upper and lower boundaries for applications to be perceived as acceptable by the user. By exceeding these boundaries, services will be considered unacceptable. Our goal has been an attempt to classify and map VR services to these classes.

III. PROPOSED MAPPING OF VR SERVICES TO UMTS QOS CLASSES

Based on referenced work and existing standards we propose a classification of VR services defining five distinct service classes. The main distinguishing factors used to classify the services are delay sensitivity and degree of interactivity. Shown in Table 1 is a mapping of the proposed classes to UMTS QoS classes with a mapping of VR service requirements to performance parameters and target values given for the UMTS classes.

Highly interactive VR applications (e.g. CVEs, simulations, network games) are comparable to UMTS conversational class services in terms of strict delay requirements. Due to the fact that network requirements depend on the nature of the application, we find it important to distinguish between *hard real time* and *soft real time interactive VR* services.

The given network requirements for streaming media components in VEs correspond to standard requirements for audio and video, with exact values for network parameters depending on the specific type of codec. When spatial audio/video is involved, requirements also depend on the relative position of the media in the virtual world.

We map VR services characterized by an end user interacting with remote equipment to the UMTS interactive QoS class. Often times we are looking at services with requirements comparable to classical Web browsing, including VEs where the user navigates from one virtual space to another, or requests downloads of additional VE objects. The *non-real time best effort VR* class includes applications that are not time dependent. This refers to single-user cases when the user simply downloads a VE and performs all other interactions with the environment locally.

Outlined in Table 1 are the target QoS values representing end-to-end performance between communicating entities. In order to realize a certain network QoS, a bearer service needs to be defined with characteristics and functionality from the source to the destination of a service. Due to the fact that a specific VR application may consist of a collection of VE objects containing various media types, a mapping of application requirements to network parameters requires specifying the requirements of each media type, along with any necessary synchronization. The "media profiles", contained within the VR service profile, represent network requirements of the different media components and need to be mapped to corresponding service component attributes.

TABLE 1. MAPPING OF VR SERVICE CLASSES TO UMTS QOS CLASSES

UMTS QoS classes	**VR Service classes**	**Media types**	**Example VR services**	**Example VR applications**	**Classification parameters**			
					Degree of sym.	**Delay**	**Delay variation**	**Information loss**
Conversational class	Hard real time interactive VR	Data, graphics, audio, video, text	Distributed simulations	Tightly coupled flight simulation	Two way	End-to-end one way delay < 100 ms	50 ms	< 3% packet loss ratio (PLR)
			Tightly coupled CVE	Collaborative engineering and design; Tele-surgery	Two way	End-to-end one way delay < 100 ms	N.A.	Zero
			Virtual conferencing	Virtual chat space: 3D graphics, audio communication	Two way	End-to-end one way delay < 150 ms	< 1 ms (for conversational voice)	< 3 % PLR
			Multi-user interactive games	First person shooter game	Two way	End-to-end one way delay < 100 ms	N.A.	Zero
	Soft real time interactive VR	Data, graphics, text	Loosely coupled CVE	Virtual multi-user shopping center	Two way	End-to-end one way delay < 400 ms	N.A.	Zero
			Multi-user interactive games	Real time strategy game	Two way	End-to-end one way delay < 500 ms	N.A.	Zero
Streaming class	VR with integrated real time streaming media (one-way)	Data, graphics, audio, video, text	VE with integrated video/audio	Virtual movie theater	Primarily one-way	Start up delay < 10 s Lip-synch. +/-80 ms	< 2 s *	< 2% PLR (video) < 1% PLR (audio)
			Virtual humans on the Internet	Virtual newscaster: streaming audio, video, and animation	Primarily one-way	Start up delay < 10 s Lip-synch. (animation and audio) +/-80 ms	< 2 s *	< 2% PLR (video) < 1% PLR (audio) < Zero % PLR (animation)
			VE with integrated audio	VE with background music	Primarily one-way	Start up delay < 10 s	< 2 s *	< 1% PLR
Interactive class	Non-real time interactive VR	Data, graphics, text	Virtual place simulation on the Internet	Virtual city tour – user navigates through multiple virtual spaces	Primarily one-way	One way delay < 4 s/new space	N.A.	Zero
			Data base retrieval	3D dynamic data visualization – user interacts with data	Primarily one-way	One way delay < 15 s/page	N.A.	Zero
Background class	Non-real time best effort VR	Data, graphics, text	Background download of VE	VE requiring only initial download – all subsequent interactions/object manipulations occur locally	Primarily one-way	No special requirements – download time depends on size of VE components	N.A.	Zero

*Values given by 3GPP specification and corresponding to one-way audio/video streaming services. Requirements on delay variation in terms of human perception are very strict, but due to the fact that the stream is aligned at the receiving end, the highest acceptable delay variation is given by the capability of the time alignment function of the application.

IV. CASE STUDIES: VR SERVICE MODEL AND MEASUREMENTS

Case studies have been conducted to determine the end-to-end requirements of a number of heterogeneous VR applications, which serve to verify the proposed service classification and mapping.

We consider VR services in the context of a general model proposed for VR service design and delivery that enables transparent user access. The parameters of a VR "service profile" are matched with restrictive parameters of the "client profile" to achieve service adaptation resulting in the "highest achievable" quality, from the point of view of the user and the VR service provider. Considered test cases are classified as belonging to VR service classes based on measurements of network parameters identified in the service profile. Where possible, the adaptation of service parameters enabled us to consider different service versions with variations in application/network level QoS requirements.

A. Service adaptation model

The model [12], shown in Figure 2, represents an extension of a general VR framework [10] and builds on existing standards and techniques for multimedia (trans)coding, compression and networking, as well as 3D graphics optimization, in order to achieve service adaptation. It consists of three main components:

- *Client*
- *Access Server*
- *Application Server*

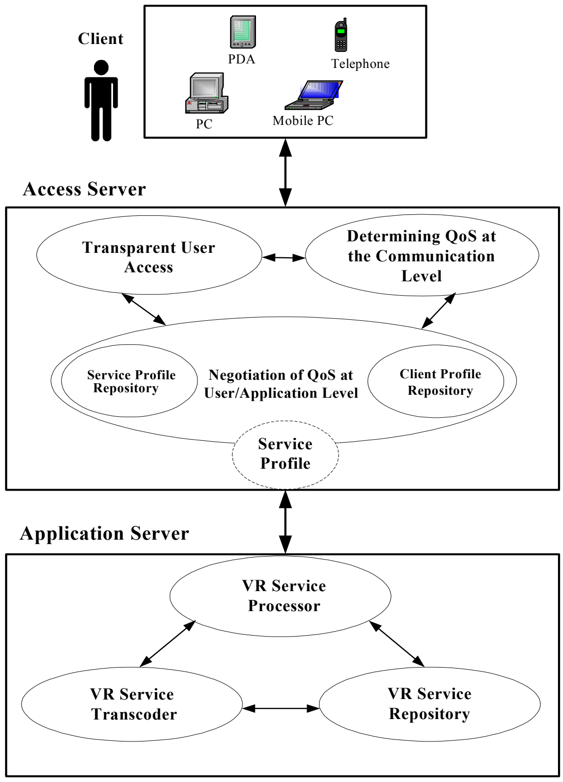

Figure 2. Service Adaptation Model

The term "client" denotes a particular combination of terminal hardware, operating system, and client software application. Upon the user's request, the client contacts the *Access Server*. The *Access Server* identifies the client characteristics, determines and adapts communication and application QoS. This is based on matching parameters of the client profile with parameters of the VR service profile (located in the service profile repository). An optimal VR service profile is generated for a particular client-service combination and communicated to the *Application Server*, which is responsible for retrieving the VR service from the *VR service repository* and transforming the service as needed based on the generated VR service profile. Adapted content is returned to the *Access Server* and passed back to the client. It may be noted that the QoS matching and adaptation remains completely transparent to the client.

The generic VR service profile (Figure 3) contains four sets of parameters: general service information, processing requirements, network requirements, and special options (used for parameters that need to be more precisely specified).

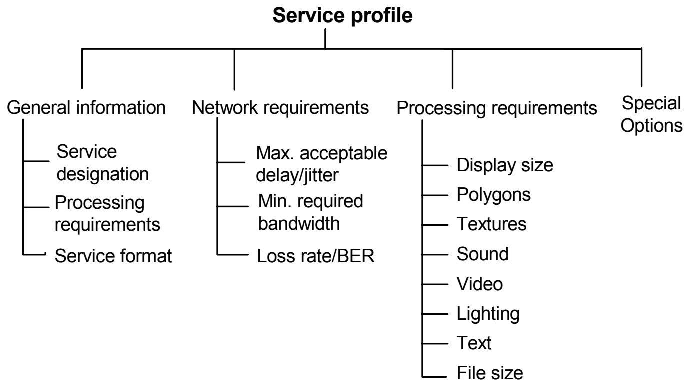

Figure 3. Generic VR service profile

The indicated service parameters address QoS at the application and network level and can be related to parameters characterizing a bearer service as defined by 3GPP. According to specifications [1], an application will specify its network QoS by negotiating a bearer service with:

- a specific traffic type: guaranteed/constant bit rate, non-guaranteed/dynamically variable bit rate, real time dynamically variable bit rate with minimum guaranteed bit rate
- traffic characteristics: point-to-point, point-to-multipoint
- maximum transfer delay
- delay variation
- bit error rate
- data rate

Exact bearer service attributes and their relevancy for each QoS class can be found in [2]. Rules for deriving these attributes are discussed in [3] and are based on service description parameters such as those of the proposed service profile. Exact mapping is outside the scope of this paper.

B. Measurements

Measurements were conducted in a laboratory LAN involving the mentioned service parameters to determine end-to-end QoS requirements. Results serve to classify test case applications as belonging to proposed VR service classes.

Requirements are determined by testing the effects of various network conditions on user perceived quality. Under certain conditions (i.e. an end user with limited bandwidth due to access network capabilities), a service may not achieve its intended functionality and may therefore be considered unacceptable.

Our testbed LAN consisted of four PCs and one WS connected by a 10 Mbit/s Ethernet LAN. The *NIST Net* network emulator tool (http://snad.ncsl.nist.gov/itg/nistnet/) was installed and used to emulate numerous network conditions. Performance scenarios that can be emulated include bandwidth limitations, tunable delay distributions, congestion and background loss, and packet reordering/duplication.

Delay in the LAN was measured (using ping) to be less than 1ms and packet loss 0%. The testbed configuration included the following hardware and software:

- PC 1 – Pentium IV (1.6 GHz, 512 MB RAM), Linux 2.4.17 (RedHat 7.2) OS; NIST Net network emulation package version 2.0.10.
- PC 2 – Pentium III (750 MHz, 256 MB RAM), Windows 2000 Professional OS; Cortona 3.0 VRML plug-in; Blaxxun Contact multi-user 3D plug-in; Java Media Framework 2.1.1.
- PC 3 – Pentium IV (1.6 GHz, 512 MB RAM), Windows XP Professional OS; Apache httpd Server v.1.2.17; Savant Web Server; Alicebot.net Server 4.0.
- PC 4 – Pentium III (600 MHz, 256 MB RAM), Windows 2000 Professional OS; Ethereal (network protocol analyzer) version 0.9.7; Blaxxun Contact.
- WS 1 – Sun Sparc Ultra 5, Sparc v9 (270 MHz, 256 MB RAM), Solaris 8.0 OS; Blaxxun Virtual Worlds Platform 5.1.

Case studies were organized into two sets of measurements:

- *Single user VEs*: Virtual Phone Gallery, conversational virtual character Demy
- *Multi-user VEs*: Virtual Audio Chat, multiplayer Medal of Honor, Blaxxun virtual community.

1) Single user virtual environments

In the case of single user VEs, only interactions between the user and the service were observed, rather than interactions between multiple users. Our research was oriented towards requirements for VEs on the Internet, where one of the key issues was time necessary for scene download. User interactions may trigger additional network traffic by requesting one-way audio/video streaming, or additional file download. Two test cases were addressed.

In the first case, a user accesses a Virtual Phone Gallery, developed using the Virtual Reality Modeling Language (VRML), from a WWW server. The only generated network traffic is HTTP/TCP traffic during scene download. The VR service was implemented in three versions differing in complexity: *high quality* (HQ), *low quality* (LQ), and *handheld*. The service profile belonging to the most complex version of the service (HQ) identified it as having a display size of 1024x768 pixels, 10000 polygons, lighting complexity value 18, texture size of 298 kB, texture color depth 24 bits, audio clip size of 32.5 kB and file size of 1524 kB. The simpler version (LQ) had the same display size, 1970 polygons, lighting complexity value 1, texture size of 26.7 kB, texture color depth 24 bits, file size of 252 kB and no audio. A third version of the service (Handheld) was implemented in which the display size of the LQ version was modified to 240x320 pixels. The NIST Net tool was used to determine the effects of various bandwidth availabilities and delays on download time. Loss probability was kept at 0%. Results are shown in Figure 4 and Figure 5. Download time for the *handheld* version corresponds to values measured for the *LQ* version. Values for bandwidth and delay correspond to one-way values and were set in both directions (symmetric links assumed for simplicity).

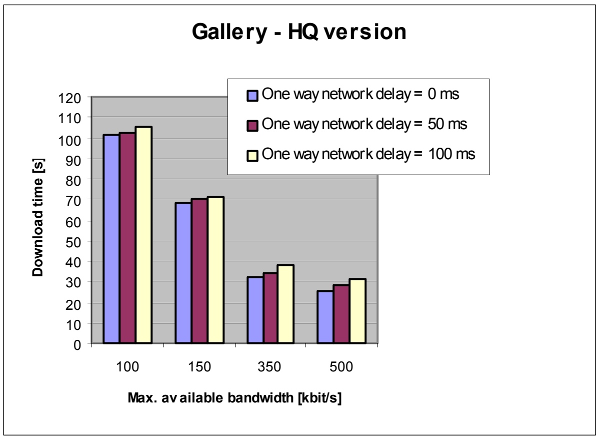

Figure 4. Download time for HQ *Virtual Phone Gallery*

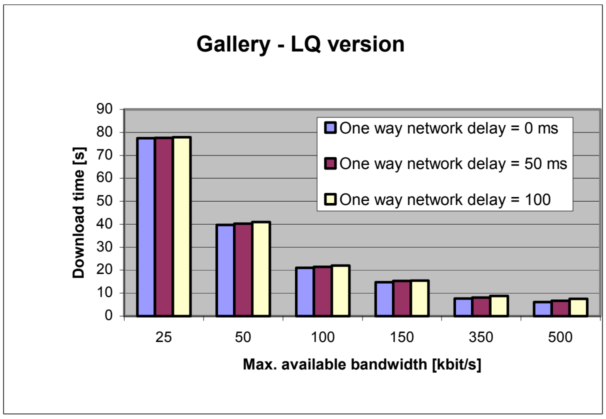

Figure 5. Download time for LQ *Virtual Phone Gallery*

The idea is to determine the network requirements of the service such that download time is considered acceptable by the end user. Depending on end user network/terminal capabilities and the time considered acceptable for download, the appropriate service version is returned.

In the second test case, we look at a conversational virtual character designed for the Web that is capable of having a meaningful conversation with a user who types in the input [16]. The artificial intelligence of the character is based on the latest Java implementation of ALICE (http://www.alicebot.org). Answers are based on "knowledge" contained in an Artificial Intelligence Markup Language (AIML) file. The animated virtual character (Java applet) can be controlled by JavaScript and instructed to talk. Speech is stored on the HTTP server in the form of audio (.au) files and MPEG-4 facial animation files (.fba). Measurements were performed to determine the time necessary for the virtual character to respond to a question asked by the user (Time to Answer - TTA). This involved the generation of an *.au* file and an *.fba* file on the server side that are streamed across the Web to the end-user. Measurements of TTA can be found in [16] where tests were performed in a best effort Internet environment. Our goal was to perform tests in an emulated network environment with control over network parameters. The server configuration used during tests corresponds to PC 3. TTA was measured for responses generated based on two different questions, Q1 “Hello” (response: “Hi there!”, .fba file size 90 B, .au file size 1.83 KB), and Q2 “What do you do?” (response: “Human, I talk to people on the Web. What do you do?”, .fba file size 281 B, .au file size 2.25 KB). The length of the generated response affects TTA (Figure 6). Values for bandwidth and delay correspond to one-way values set in both directions.

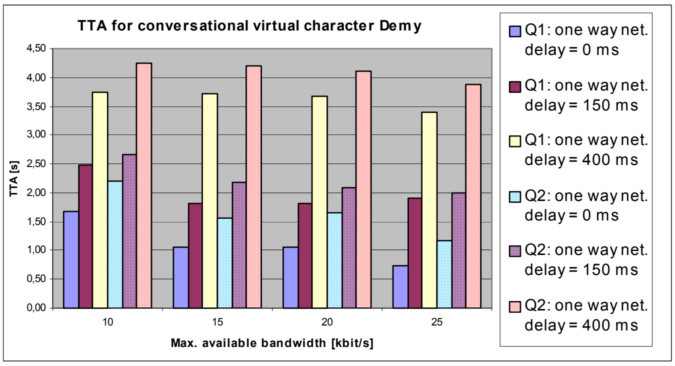

Figure 6. Measurements of TTA for *Demy*

The user perceived quality of virtual characters depends largely on the purpose of the application, whether it be entertainment, commerce, education, or personal communications. The issue is to determine the requirements of the application in order for it to be functional and attractive. In our case, long response delays during the chat reduce the user feeling of ”real-time” interactivity. For practical implementation, experience shows that TTA within the range of 1-2 s would be acceptable. Results show the necessary network conditions for achieving such quality. Based on user terminal capabilities, different implementations of the virtual character may be returned varying in complexity (i.e. the number of polygons used) and frame rate. Response times, however, remain unchanged.

2) *Multi-user virtual environments*

In multi-user VEs, multiple users from geographically distributed locations can communicate, collaborate or interact with each other and the environment. Such services often require large bandwidth and low latency. In order to consider some of the requirements of such services, measurements were performed using three applications.

The first application considered is *Virtual Audio Chat* (VAC) that enables real time audio communication over the Internet between a multiple numbers of users [11]. The user interface includes a VRML model of a mobile phone. By way of user interactions, a user can enter onto the phone the IP address and port number of another user (or a multicast address/port in the case of a group session) with whom he wishes to communicate. A Java applet opens a Real Time Transport Protocol (RTP) based audio streaming session using the entered address and port, and waits for the other user to join in an analogous manner. Network traffic corresponding to this application includes HTTP/TCP traffic during initial download, and afterwards a continuous RTP/UDP stream for the duration of the audio chat. RTP works in combination with the RTP control protocol (RTCP) that monitors data delivery and provides control over data transport. Network requirements depend primarily on the codec being used. Based on network requirements indicated in the client profile, the application is returned to the user implementing one of three different codecs: PCM (64 kbit/s), GSM (13 kbit/s), or G.723.1 (5.3 and 6.3 kbit/s). RTP throughput for the mentioned codecs (taking into account all headers) is shown in Figure 7.

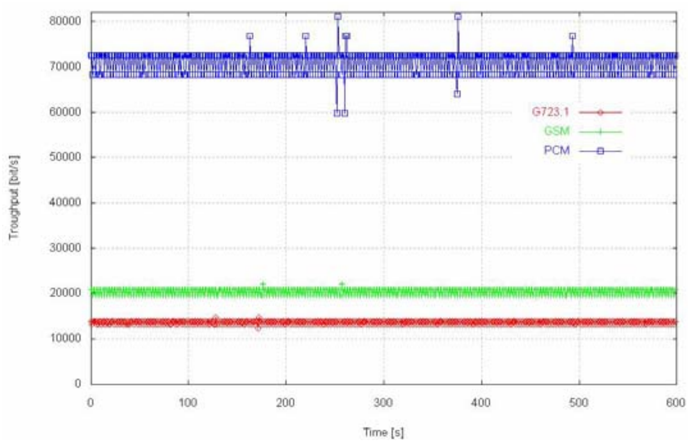

Figure 7. RTP packet throughput

Requirements concerning end-to-end delay, delay jitter, and loss are based on accepted values specified in standards concerning conversational voice [5][6]. Measurements of end-to-end delay conducted to determine delay imposed by the application itself (network delay less than 1ms) found it to be approximately 500 ms. Although this exceeds target values given in standards, we found users were able to conduct a conversation without greater difficulty.

In the second test, users took part in a multiplayer FPS computer game known as *Medal of Honor* (http://www.ea.com/eagames/official/moh_alliedassault/home.jsp), characterized by high quality 3D worlds that utilize the powerful Quake III engine. Measurements were performed to determine effects of bandwidth limitations, delay, and jitter on user perceived quality. One user hosted the game while the other user joined in. Measured traffic (Figure 8) shows that the joining player generated nearly twice as much traffic as the host player. The dynamic nature of generated traffic corresponds to dynamic user interactions.

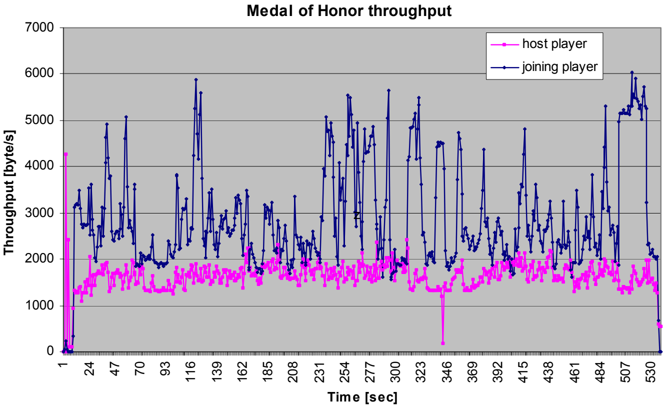

Figure 8. Measured throughput for *Medal of Honor*

Measurements showed that a minimum of approximately 2200 Bytes/s in both directions is necessary in order to play the game. Delay effects were measured by setting delays in both directions and asking players to comment on quality of fair play. Delay remained unnoticed by the host player, while at a one-way delay of 100 ms (200 ms round trip time), the joining player reported noticeable delay. Jitter effects were reported to be only slightly noticeable at a set jitter of 40 ms in each direction. Due to the fact that the service was available in only one version, a user could choose whether to join or not.

The third multi-user VE test case involved a shared virtual environment built upon Blaxxun's Virtual Worlds Platform that allows multiple users to meet and interact. A client-server architecture is used to distribute updates among members in real time using the UDP protocol. In multi-user interactive VEs, changes resulting from a user's actions need to be made visible to other users in a consistent manner to achieve "real-time" interactivity, making delay a key factor determining user perceived quality. In the case of avatar gestures, only packets carrying information to trigger the action are sent. Due to low bandwidth utilization, after initial download there was no need to test bandwidth limitations. Measurements were conducted to determine delay allowed in order to maintain acceptable user perceived quality of real time interactivity. End-to-end delay was measured from the moment when one user triggered a waving gesture until that gesture was made visible to the other user (measured using Video Blaster WebCam 3 USB). Delay values up to 300 ms in each direction remained practically unnoticed by users. At 400 ms delay in each direction became noticeable. Specific values of acceptable delay depend on the nature of the interactive VR application. In this particular scenario, involving communication through gestures, delay is more tolerable than in multi-user interactive games [17] or military simulations simulating combat [14].

V. Discussion of Results

A mapping of applications addressed as case studies to proposed VR service classes is based on achieved test results. The mapping is given in Table 2. When using mobile devices or computers without VR-specific input and output devices, a user's sense of immersion is limited by device capabilities, which also affect the mapping and user perceived QoS. For example, when a user is served an adapted, lower quality service version as opposed to a higher quality version, the QoS mapping could change from conversational to interactive class.

Due to strict delay requirements and high interactivity, we classify the multi-user shooting game as a hard real time multi-user interactive VR application. Tests performed using the Blaxxun virtual community showed that delay became noticeable at approximately 400 ms (end-to-end). Stricter delay limitations were not necessary with participants interacting using only gestures. This application was therefore classified as belonging to the soft real time multi-user interactive VR service class.

In the case of Virtual Audio Chat, different codecs are enabled during the chat depending on end user capabilities. Measurements were performed with users engaging in a conversation to determine delay requirements. Users reported that conversation was possible without difficulty as long as end-to-end one way delay was kept less than 500 ms. We classify this application as soft real time multi-user interactive VR. Achieving conversational quality comparable to normal telephony speech would require stricter requirements on delay and jitter.

The conversational virtual character is classified within the interactive VR service class due to the interactive nature of the chat. In addition, *.fba* and *.au* files are streamed from the server, however the client player waits for complete file download prior to playing the response.

The virtual mobile phone gallery, implemented in the HQ version, may be considered as belonging to the interactive VR service class due to the fact that each mobile phone model located in the gallery is downloaded only when the user falls within a certain range of that particular model. Thus, a change in the user's viewpoint may cause additional download. Requirements on delay are therefore necessary in order to prevent deterioration of navigational quality. The measurements presented in Section III correspond to download times for the entire gallery with all components, thus equivalent to a user navigating through the entire gallery upon download. The LQ gallery version downloads the entire gallery at once and is mapped to the download/best effort VR service class, with no special requirements on delay. A data rate of approximately 70 kbit/s is shown as needed in order for download time to be less than 30 s. However, this may be considered a soft requirement since a user may be willing to wait for maybe 30 seconds or one minute, before locally navigating through the gallery and examining the mobile phones.

TABLE 2. MAPPING OF TEST CASE APPLICATIONS TO VR SERVICE CLASSES

Example prototype applications	Corresponding VR Service class	Data rate	Delay	Delay variation	Information loss
Multi-user shooting game	Hard real time multi-user interactive VR	Min ≈ 20 kbit/s required	One way delay < 100 ms	< 30 ms jitter	Zero
Blaxxun multi-user virtual community (communication only using gestures` and text box)	Soft real time multi-user interactive VR	< 1 KB	One way delay < 400 ms	N.A.	Zero
Virtual audio chat	Soft real time multi-user interactive VR	Min. 13,6 kbit/s required (G.723.1 codec)	One way delay < 500 ms	< 20 ms jitter	< 5 % packet loss
Conversational virtual character on the Internet	Single user interactive VR	Audio ≈ 10 kbit/s Animation MPEG-4 FBA ≈ 2.4 kbit/s	Response time < 2 s	N.A.	Zero
Virtual mobile phone gallery (High quality version)	Single user interactive VR	380 kbit/s	<30 s for complete download (<5 s/ phone)	N.A.	Zero
Virtual mobile phone gallery (Low quality version)	Download and best effort VR	70 kbit/s	<30 s for download	N.A.	Zero

VI. SUMMARY AND CONCLUSIONS

In this paper we have addressed end-to-end QoS support for VR services in the context of the UMTS QoS framework specified by 3GPP. We have proposed a classification of VR services based on delivery requirements and degree of interactivity that maps to existing UMTS QoS classes. The mapping is based on matching VR service requirements to performance parameters and target values defined for UMTS applications. A number of test cases have been presented involving heterogeneous VR applications that serve to verify the proposed service classification and mapping. Key VR service parameters affecting requirements are identified using as a reference a general model for VR service design and delivery. Measurements of network parameters serve to determine the end-to-end QoS requirements of the considered applications, which are in turn mapped to proposed VR service classes.

We have demonstrated that the end-to-end QoS requirements specified for UMTS services have met the requirements of the proposed VR service model in an emulated network environment. Future steps will focus on determining support for VR services in an actual UMTS network.

REFERENCES

[1] 3GPP TS 22.105, “Services and Service Capabilities”, Release 6, September 2002.

[2] 3GPP TS 23.107, “QoS Concept and Architecture”, Release 5, September 2002.

[3] 3GPP TS 29.208, “End-to-end Quality of service signaling flows”, Release 5, September 2002.

[4] ITU-T Recommendation F.700, “A Framework Recommendation for Multimedia Services”, November 2000.

[5] ITU-T Recommendation G.1010, “End-User Multimedia QoS Categories”, November 2001.

[6] ITU-T Recommendation G.114, “General Recommendations on the transmission quality for an entire international telephone connection: One-way transmission time”, May 2000.

[7] C. Greenhalgh, S. Benford, G. Reynard, “QoS architecture for Collaborative Virtual Environments”, Proc. ACM Multimedia ’99, Orlando, Florida. ACM Press.

[8] V.D. Lehner, T.A. DeFanti, “Distributed Virtual Reality: Supporting Remote Collaboration in Vehicle Design”, IEEE Computer Graphics and Applications, Vol. 17, No. 2, March-April 1997, pp.13-17.

[9] M. R. Macedonia, M. J. Zyda, “A Taxonomy for Networked Virtual Environments”, IEEE Multimedia, Vol. 4, No. 1, Jan-March 1997, pp.48-56.

[10] 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), August 2002.

[11] M. Matijasevic, L. Skorin-Kapov. “Design and Evaluation of a Multi-user Virtual Audio Chat”, Future Generation Computer Systems, Vol.19, No. 2, 2003, pp. 229-239.

[12] D. Mikic, D. Vilendecic, L. Skorin-Kapov, M. Matijasevic. “Modeling and Design of an Adaptive Virtual Reality Service”, Proc. of 27th IEEE Conference on Local Computer Networks, Tampa, Florida, Nov. 6-8, 2002, pp. 315-324.

[13] K.S. Park and R.V.Kenyon, “Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment”, IEEE Intl. Conference on VR (VR ‘99), Houston, Texas, March 1999.

[14] J. Pullen, D. Wood, “Networking Technology and DIS,” Proceedings of the IEEE, 83(8), August 1995.

[15] S. Shirmohammadi, N.Georganas, “An End-to-End Communication Architecture for Collaborative Virtual Environments”, Computer Networks, Vol.35. No. 2-3, pp.351-367, 2001.

[16] I. S. Pandzic, R. Forchheimer (editors). MPEG-4 Facial Animation Framework for the Web and Mobile Platforms in “MPEG-4 Facial Animation - The standard, implementations and applications". John Wiley & Sons, 2002, ISBN 0-470-84465-5.

[17] J. Smed, T. Kaukoranta, H. Hakonen, “Aspects of Networking in Multiplayer Computer Games”, Proc. Int’l Conf. on Application And Development of Computer Games in the 21st Century, 2001.