Dynamic QoS Negotiation and Adaptation for Networked Virtual Reality Services

Lea Skorin-Kapov
Ericsson Nikola Tesla
Research and Development Center
Krapinska 45, Zagreb, Croatia
lea.skorin-kapov@ericsson.com

Maja Matijasevic
University of Zagreb
Faculty of Electrical Engineering and Computing
Unska 3, Zagreb, Croatia
maja.matijasevic@fer.hr

Abstract

Providing seamless and acceptable Quality of Service (QoS) support for multimedia services in heterogeneous environments implies an adaptive QoS system, capable not only of establishing end-to-end QoS support at session setup, but also of reacting to changes dynamically occurring during the session. In this paper, we propose a model for the dynamic negotiation and adaptation of QoS parameters for Networked Virtual Reality (NVR) services, considering such services to be among the most complex type of multimedia. In addition, we propose generic client and service profiles used in the negotiation process. While the model is independent of the particular service and network scenario, certain elements are influenced by QoS requirements specified for the Universal Mobile Telecommunications System (UMTS). The model is implemented in a laboratory testbed and a case study is described to verify functionality.

1. Introduction

With advances in computer and networking technologies comes the challenge of addressing heterogeneous environments for both service users and service providers offering new multimedia applications ”anywhere-anytime” [2]. Considering the next generation network (NGN) architecture, the necessary assumptions of user, terminal, and service mobility impose a tough task on providing adequate Quality of Service (QoS) support. Providing end-users with seamless and acceptable QoS support implies an adaptive QoS system capable not only of establishing necessary QoS support at session setup, but of reacting to dynamic changes during the session. These may include changes in the characteristics and capabilities of the end user, changes in the network (e.g. dynamic resource availability/cost), and dynamic service requirements (e.g. addition of new media components to an active session). The provisioning of end-to-end QoS is based on successful QoS negotiation between involved parties in order to agree on a common and feasible set of QoS parameters.

In this paper we focus on issues related to the negotiation and dynamic adaptation of QoS parameters for Networked Virtual Reality (NVR), considering such services to be a good representative of advanced multimedia services for the NGN. Characterized by 3D graphics and integrated multimedia components, NVR services represent interactive applications that often times go a step beyond traditional multimedia. An example of NVR may be a networked 3D game, a virtual telemedicine application, or distance education application. By considering the NVR service to be comprised of a collection of virtual world objects (including 3D graphics and integrated multimedia), communication requirements may be expressed as a combination of the requirements of particular objects mapped down to transport level QoS parameters. In addition to standard QoS parameters such as bandwidth and delay, factors such as a user’s level of interest in an object inside the virtual 3D world need to be taken into account.

In this paper, we propose a model providing dynamic QoS support for NVR services. Focus is on the overall process of end-to-end QoS negotiation and service adaptation from when a user accesses a service until service termination. The main issues to be taken into account have been identified as: (1) user terminal and access network constraints; (2) way(s) of expressing user preferences in terms of application components (media elements); (3) dynamic resource availability and cost; and, (4) mapping of user/application requirements to transport QoS parameters. In addition, we propose a generic client profile as a way of expressing client capabilities and preferences, and a generic service profile to specify dynamic service requirements. While the proposed model is independent of the particular service and network scenario, certain elements are influenced by QoS requirements specified for the Universal Mobile Telecommunications System (UMTS) architecture.

The paper is organized as follows. The model functionality is presented in Section II. Section III describes model implementation and shows results using a prototype NVR service as a case study. In Section IV, we compare our model with related work. Section V concludes the paper.

2. Proposed Dynamic QoS Adaptation Model

The proposed model targets the overall process of end-to-end QoS negotiation/renegotiation and service adaptation. The model, shown in Figure 1 [9], is composed of a number of functional entities grouped logically into three components: *Client*, *Access and Control*, and *Application Server*.

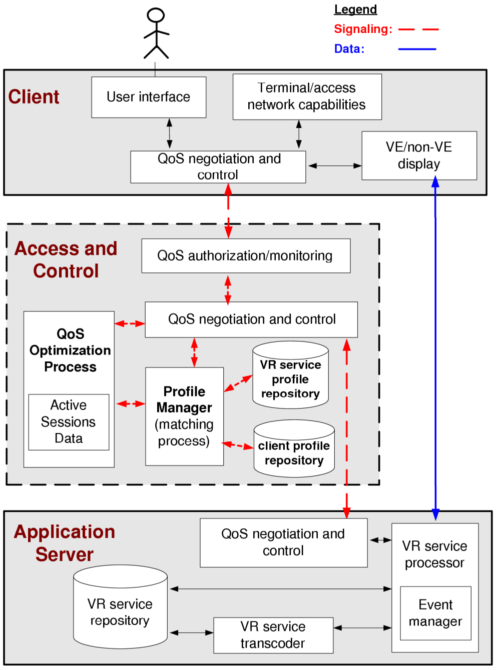


Figure 1. Model for dynamic negotiation and adaptation of QoS.

Upon a user's request for a particular service, the *Client* sends a request to the *Access and Control* entities, specifying user preferences and client capabilities in a client profile. The profile may be sent by the client or referenced in a *client profile repository*. The *Access and Control* entities are responsible for identifying the client, authorizing, and negotiating QoS parameters for the requested service.

Negotiated and authorized parameters serve as input to an optimization process designed to dynamically calculate the resource allocation and application operating point. After calculation, reservation mechanisms are invoked to reserve network resources. Upon successful reservation, adapted application content is retrieved from the *Application Server* and returned to the user.

We assume that all session related media and control information is handled by the *Access and Control* and *Application Server* entities to enable the dynamic calculation of an optimal application operating point. Optimization and renegotiation/adaptation procedures are invoked during the course of the service lifetime in response to changes occurring in: resource availability and/or resource cost; the client profile; and VR service requirements.

2.1. Access and Control

The *Access and Control* component represents a logical grouping of service control functionalities, with it being likely in an actual network scenario that proposed functionalities be distributed among different entities in the network.

2.1.1 Client profile determination

A user's initial service request is constructed by way of a user interface enabling a user to request a particular service and configure corresponding preferences. Options include enabling a user to specify desired network quality (e.g. high, medium, low) of media components, preferences such as "audio has priority over video", a constrained budget, minimum acceptable framerate, and maximum acceptable download time. These preferences, together with terminal hardware, terminal software, and access network characteristics are incorporated into a client profile. The proposed generic client profile is shown in Figure 2. The specification of certain parameters was influenced by parameters specified in existing profiles conforming to the Composite Capabilities/Preferences Profile (CC/PP) framework proposed by the WWW Consortium.

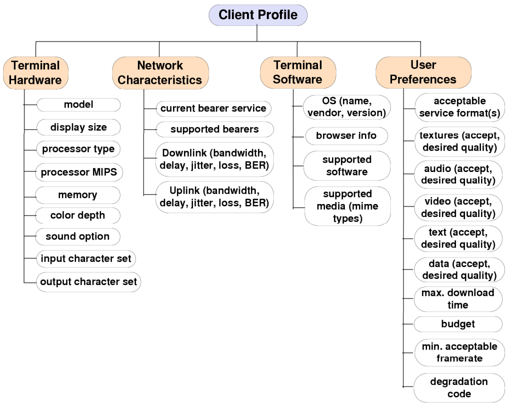


Figure 2. Generic client profile.

2.1.2 Profile Manager

The client request is received by the *QoS Negotiation and Control* (QNC) module (of the *Access and Control* component) and includes or references the client profile. The profile, or a reference to a profile in a *client profile repository*, is passed to the *Profile Manager* (PM) module. The PM matches the restrictive parameters of the client profile with parameters of the VR service profile(s) corresponding to the requested service in order to determine feasible service configurations.

The *VR service profile repository* is a repository of profiles specifying supported configurations (versions) of VR services. For example, an application may be implemented in two ways: with media streaming (configuration 1), and without media streaming (configuration 2). In that case, two profiles for the application would be stored in the repository. We specify a generic service profile as shown in Figure 3.

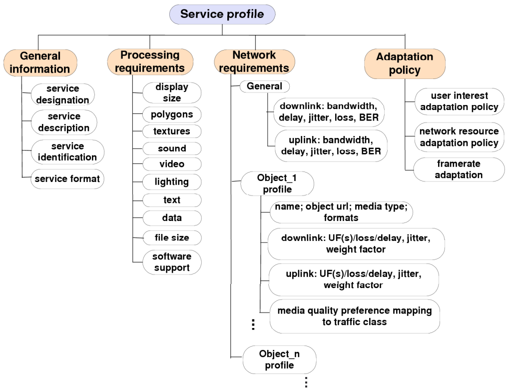
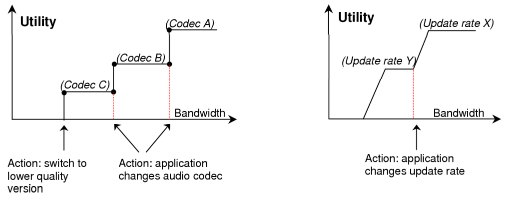


Figure 3. Generic service profile.

Aside from specifying standard parameters related to hardware and software configurations, we present an approach for specifying dynamic network requirements integrating object utility functions (UFs) and the notion of their relative importance. When specifying the requirements of NVR services, we use as a reference the general VR framework given in [4], showing the relationship between interactions and objects at the application level and traffic parameters needed for QoS. A more in-depth discussion of the parameters determining the end-to-end requirements of NVR services, with a mapping to existing QoS parameters and classes specified for UMTS, is given in [7].

If we look at the network requirement parameters, the first subset corresponds to general requirements, i.e. the minimum requirements that need to be supported by the access network for the service to be feasible. Network requirements for an object are specified in the form of UFs (showing dependence of user value on bandwidth allocation and specified for discrete values of delay and loss), with utility values ranging from 0 (minimum) to 1 (maximum requirements). A user's interest in objects may be calculated based on viewpoint/position updates, user task, and/or user defined preferences. We refer to this as calculated Level of Interest (LOI), which may be used to assign a weight factor (WF) to corresponding UFs. Examples of UFs and actions (regarding application configuration) when certain bandwidth thresholds are reached are shown in Figure 4.



(a) UF for audio streaming object

(b) UF for shared 3D graphical object

Figure 4. Example UFs for virtual objects.

The last set of profile parameters specifies application specific *Adaptation Policy*, or the actions to be taken when certain parameter values are reached, as follows:

- **User interest adaptation policy**: actions to be taken as a result of changes in a user's LOI (e.g. set WF).
- **Network resource adaptation policy**: actions to be taken as a result of changes in network resource availability (e.g. switch codec, degrade/upgrade service).
- **Frame rate adaptation policy**: the order in which to degrade the quality of objects to increase frame rate.

Certain parameters of the service profile relating to network requirements are updated dynamically over the course of the service lifetime (e.g. media streaming object is started/stopped; a new user joins a shared 3D environment; certain weight factors have changed). Different parameter set(s) (*object profiles*) may be considered throughout the service lifetime to calculate necessary resource allocation, depending on which objects are currently ”active”.

The matching process conducted by the PM to determine feasible service configurations is based on comparing client and service profiles. A service configuration is considered *feasible* when: a client's terminal capabilities support service processing requirements; the client's access network supports the minimum network requirements for all virtual world objects; and client preferences in terms of acceptable media components and allowed download time can be met.

After the matching process, the PM extracts a set of potential session parameters (e.g. media formats, codec types, etc) from those service configurations that are feasible. These are passed to the QNC module, which sends a session offer to the *Client*. The *Client* receives the offer and may accept/deny/modify offered parameters. A message is returned indicating the subset of offered parameters agreed to by the *Client*. Network entities authorize resources based on the agreed parameter subset. For illustration purposes, we assume an authorization process based on the process specified by 3GPP for multimedia services in UMTS. The authorization includes limits on data rates and traffic classes for uplink/downlink flows and is determined based on QoS policy and admission control mechanisms in the network.

The returned session parameter subset and authorization is passed back to the PM, which then orders the feasible service configurations based on achievable user perceived quality into a so-called *degradation path* from the highest to the lowest quality configuration. At this point, certain service configurations may be eliminated from consideration due to *Client* rejection of certain offered service parameters. Establishment of a degradation path is influenced by user preferences (e.g. a user considers audio to be more valuable than video). This is used when service degradation or upgrading is necessary. Finally, the service profile corresponding to the highest quality feasible configuration is passed on to the *QoS Optimization Process* (QOP) module.

The *QOP* is responsible for calculating the optimal resource allocation and application operating point, while taking into account given constraints (described later). The goal is to maximize user perceived quality by combining the user's notion of VR object "importance" and UF based adaptation. After calculation, the *QOP* passes the final profile (via the *QNC* module) specifying the calculated service operating point and required resources to the *QNC* module of the *Application Server*. The *Application Server* initiates necessary resource reservation mechanisms and signals the final session description to the *Client*, who based on the description, also initiates necessary resource reservation.

2.2. Application Server

The *Application Server* is responsible for retrieving and adapting stored VR service content. The *QNC* module receives the final profile and initiates resource reservation based on the necessary resources specified in the profile. The final service profile is then passed to the *VR service processor*. It retrieves the service content from the *VR service repository*, which stores all components that make up the virtual world scenes, including: source code of a virtual world scene; textures applied to virtual world objects; and multimedia components (e.g. sound files, video clips).

Retrieved content is passed to the *VR service transcoder* if necessary, whose job it is to convert content format (e.g. from XML to VRML) and modify content as requested (e.g. compression, filtering). Once the content has been adapted (i.e. the calculated operating point established), the *VR service processor* returns the content to the end user.

A user's interest in virtual world objects is subject to dynamic changes. Once a significant change in interest is detected, there may be a need to recalculate the optimal application operating point and reallocate network resources to meet new QoS requirements. A change in user interest, a user interaction, or a change detected by the application cause an event to be passed from the application to the *Event Manager* (EM). The *EM* then determines the current LOI of virtual world objects. These LOIs are sent to the *QOP*, which consults the *user interest adaptation policy* (of the service profile) and updates object profiles. The process of initial NVR session establishment is shown in Figure 5.

2.3. QoS Optimization Process

In this section, we further describe the optimization process conducted by the *QOP* of the *Access and Control* component. After the highest quality service configuration has been determined by the *PM*, it remains up to the *QOP* to determine the amount of resources to be allocated and the optimal operating point, defined as the one that maximizes user perceived quality. User perceived quality is expressed as a function of transmission parameters (in the form of UFs) multiplied by corresponding WFs indicating the relative importance of virtual world objects.

The function to be maximized is a linear combination of UFs multiplied by WFs defined across all objects. Only those UFs marked as active are taken into account at a certain time. UFs are defined separately for uplink and downlink. The UF for a given object in a specified direction (uplink or downlink) is defined as a function of transmission parameters $x_{i,1}, \ldots, x_{i,n}$ (bandwidth, delay, loss, etc) and expressed as $U_i(x_{i,1}, \ldots, x_{i,n})$. W_i represents the weight factor corresponding to U_i. We consider functions $U_i, i = 1, \ldots, k$ as the UFs specified for the downlink direction, and $U_i, i = k + 1, \ldots, m$ as the UFs specified for the uplink direction. Decision variables are represented as follows: $x_{i,j}, i = 1, \ldots, m, j = 1, \ldots, n$. We define $x_{i,j}$ as the value of transmission parameter j for a virtual world object in a given direction (uplink or downlink). Variables $x_{i,j}, i = 1, \ldots, k, j = 1, \ldots, n$ represent the transmission parameters corresponding to UFs for objects in the downlink direction, and variables $x_{i,j}, i = k + 1, \ldots, m, j = 1, \ldots, n$ represent transmission parameters corresponding to UFs for objects in the uplink direction. The objective function is subject to a number of constraints:

- **Resource availability (user's network connection):** B_{userDL} and B_{userUL} represent the available downlink and uplink bandwidth at the user's network connection respectively, and $x_{i,1} = b_i$ represents the allo-

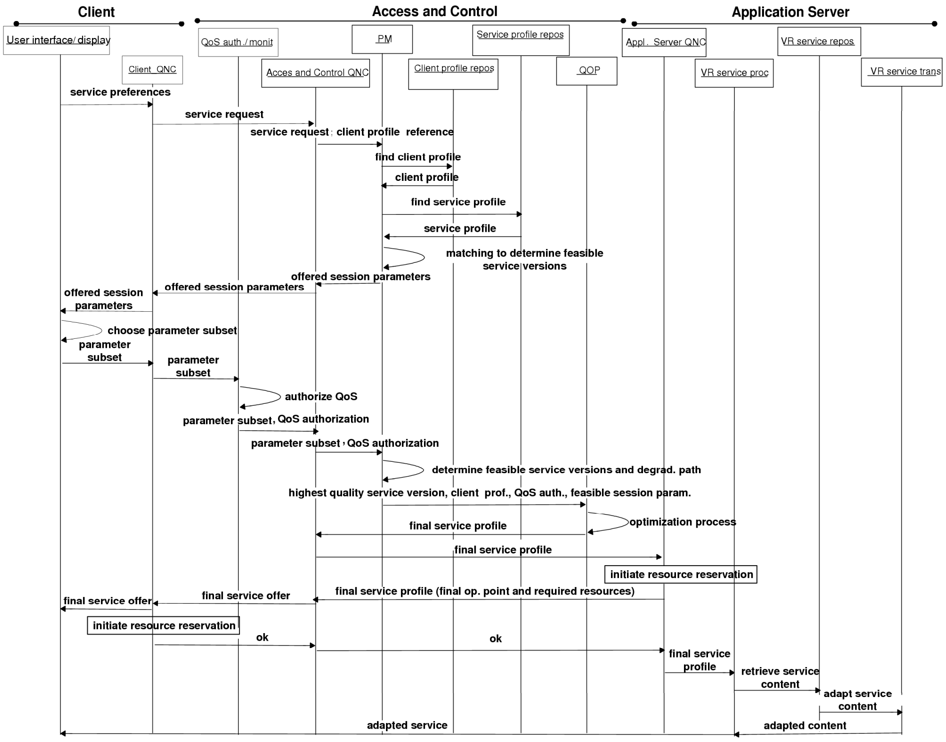


Figure 5. Sequence diagram for initial VR session establishment.

cated bandwidth corresponding to U_i.

$$\sum_{i=1}^{k} b_i \leq B_{userDL}; \sum_{i=k+1}^{m} b_i \leq B_{userUL} \tag{1}$$

- **Resource availability (authorized network resources):** $b_{iMaxAuthDL}$ and $b_{iMaxAuthUL}$ are the maximum data rates corresponding to U_i authorized by the network. Data rates are authorized per media flow and for both uplink and downlink directions.

$$b_i \leq b_{iMaxAuthDL}, i = 1, \ldots, k; \tag{2}$$

$$b_i \leq b_{iMaxAuthUL}, i = k+1, \ldots, m \tag{3}$$

For remaining transmission parameters such as delay, jitter, and loss, expected values are greater than or equal to those available and those authorized.

- **Service requirements:** $x_{i,jMin}$ and $x_{i,jMax}$ are the minimum and maximum transmission parameter requirements corresponding to U_i.

$$x_{i,jMin} \leq x_{i,j} \leq x_{i,jMax}, i = 1, \ldots, m, j = 1, \ldots, n \tag{4}$$

- **Cost:** For the purposes of this paper, we assume a price quotation (rate) received from the network and based on a per byte basis. A different price is quoted for different levels of service (QoS classes) guaranteed to the user. d_i in monetary_ unit/byte is the rate for bandwidth b_i, with the total user budget specified as D_{user} in monetary_unit/s.

$$\sum_{i=1}^{m} d_i b_i \leq D_{user} \tag{5}$$

The objective function may be written as:

$$max \sum_{i=1}^{m} [W_i U_i(x_{i,1}, \ldots, x_{i,n}) - 0.001(d_i b_i)] \tag{6}$$

We subtract the cost multiplied by a small factor so that in cases when two or more different solutions result in the same overall utility and meet all constraints (e.g. this is a possibility in the case of a linear step UF), the resulting optimal solution corresponds to the lower cost solution. Assuming that an application will be able to request QoS from the network by choosing from alternative QoS classes, we choose to specify UFs as functions of bandwidth for discrete delay and loss values corresponding to specified QoS classes. If for a particular object multiple UFs are defined at different delay and loss values that meet requirements, then all possible combinations of UFs are considered. Depending on whether UFs are linear or nonlinear, approaches based on linear or nonlinear programming may be used to calculate the solution.

As stated earlier, optimization and renegotiation or adaptation procedures are invoked during the course of the service lifetime. For example, when the network detects a decrease in available bandwidth, a trigger is sent to the *Access and Control* QNC module, which in turn invokes the *QOP* to search for a new optimal operating point and resource allocation. As long as no solution is found within given constraints, service degradation is requested (based on degradation path). Once a solution is found, a session update is sent to the *Client*, resource allocation is adapted, and the final service profile is passed to the *Application Server*.

It is clearly not realistic to assume re-calculation of the optimal operating point at every small change in constrains (e.g. available network resources, costs, terminal capabilities). Rather, certain thresholds may be established indicating modifications important enough for re-calculation, and further leading to renegotiation and adaptation.

3. Model Implementation and Case Study

3.1. Software Implementation

The model is implemented in a laboratory testbed to verify functionality and demonstrate various dynamic adaptation scenarios. We implement the *Client*, *Access and Control*, and *Application Server* as individual software entities.

In order to address model applicability in UMTS, QoS signaling is based on the exchange of SIP messages between the *Client*, *Access and Control*, and *Application Server QNC modules*. The *QoS authorization/monitoring* entity is designed to span both connectivity and control layers. It consists of a SIP proxy that forwards messages between the *Client* and *Access and Control QNC*. At the connectivity layer, a virtual channel is established based on negotiated parameters, with all content routed through this channel.

With regards to the specification of service requirements, we use a simplified approach for specifying UFs by using only linear and piecewise linear functions. This allows for the optimization problem to be formulated as a standard mixed integer linear programming problem. The GNU Linear Programming Kit (www.gnu.org) is used.

3.2. Case Study

A prototype NVR application was developed to verify proposed model functionality. Software was installed in a laboratory testbed, with each of the model components being run on a separate PC and connected via a switch on a 10 Mbps Ethernet LAN.

The Web based application is a *Virtual Automobile Gallery*, allowing a user to navigate through a virtual gallery and view images of different automobiles. The gallery is created using VRML and viewed on the *Client* using the Cortona VRML plug-in (Figure 6). Throughout the world are stands that a user can click to view a streaming automobile commercial. The application was developed in three configurations differing in display size, and availability of media: Configuration 1 offers audio and video; Configuration 2 only audio; and Configuration 3 has no streaming.

Figure 6. Virtual Automobile Gallery.

Three service profiles were created (using XML) and stored in the service profile repository. The following objects are identified: scene description (Configurations 1, 2, and 3), audio stream downlink (Configurations 1 and 2), and video stream downlink (Configuration 1 only). For each object, the bandwidth requirements are expressed through specification of UFs for discrete values of delay and loss. The functions specified in the service profiles are shown in Figure 7. Functions are hypothetical and specified for test purposes only.

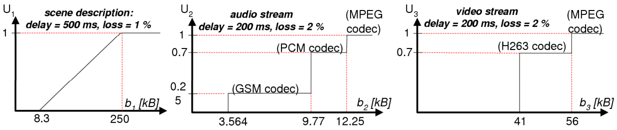


Figure 7. Hypothetical UFs.

A user requests a desired service using a graphical user interface (GUI), with different client profiles defined to depict different capabilities and preferences. The user can choose which client profile to include in the request, thus emulating different access options. In an initial scenario, a user sends a request containing the *"UMTS: high bandwidth PCHQ"* client profile. This profile specifies the capabilities of a high quality (HQ) PC, and hypothetical network characteristics corresponding to a UMTS network (downlink bandwidth = 75 kB/s, downlink delay 200 ms, downlink loss = 1%, uplink bandwidth = 30 kB/s, uplink delay 200 ms, uplink loss = 1 %). The profile also specifies a user budget of 160 monetary_unit/s, a maximum allowed download time of 20 s, and the acceptance of all media types. After the matching process, the *PM* determines that all service configurations are feasible. An offer is sent to the *Client*, who agrees to the parameters (or modifies/denies them), and sends a response back to the *Access and Control* QNC. The message is intercepted by the *QoS authorization/monitoring* entity, which adds authorization tokens for the media flows. The *PM* chooses the highest quality service configuration (1) and the optimization process is invoked. Initially, resources are only needed for scene download. After the gallery has been downloaded, the user may navigate through the 3D space. By clicking on a stand, the *Application Server* recognizes the request to start an audio/video stream, and invokes the QOP to recalculate the application operating point and resource allocation. The calculation provides the following solution: *total downlink bandwidth = 53.25, total cost = 133.125, bandwidth for audio = 12.25, bandwidth for video = 41.0.*

Based on this information and the *network resource adaptation policy*, the final operating point is determined specifying use of the MPEG codec for audio and H.263 codec for video (see Figure 7). It is clear that cost proved to be the constraining factor (total calculated cost reached maximum allowed user budget). For implementation purposes, different costs were assigned to different UMTS traffic classes, with each virtual world object being mapped to a particular class. In this case, audio/video streams belong to the streaming class (with a cost of 2.5 monetary_unit/byte), while the scene description is mapped to the background class (cost of 1 monetary_unit/byte).

Similar to the change in service requirements, changes may occur in the capabilities of the *Client*, network resource availability, or resource cost. For example, a change in the capabilities of the access network (downlink bandwidth reduced to 49 kB/s, and uplink to 14.375 kB/s) results in a new solution. Session data is updated, and a new application operating point is calculated: *total downlink bandwidth: 44.564, total cost: 111.41, bandwidth for audio = 3.564, bandwidth for video = 41.0.* The main constraining factor has now become downlink bandwidth. The operating point is determined specifying use of the GSM codec for audio and the H.263 codec for video. The effect of adaptation on the *Client* bandwidth is shown in Figure 8.

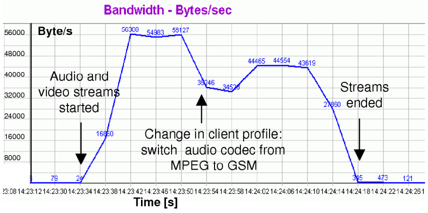


Figure 8. Client bandwidth.

4. Comparison with Related Work

Issues of QoS negotiation and adaptation have been previously addressed both in standards and literature. The authors in [10] address support for adaptive multimedia applications that adjust network requirements in response to changing prices resulting from dynamic resource availability. Adaptation is based on multi-dimensional utility functions (UF), specified for application components, showing the user perceived value of a network resource allocation. The goal is to maximize the user obtained ”value for money” corresponding to the surplus between the total utility and the total service cost. Our approach differs in that we define our goal as being to maximize overall user perceived quality, as opposed to surplus. In addition, we consider constraints related to client terminal and access network capabilities, not addressed in [10].

In [5], the authors present a QoS-aware middleware system to provide end-to-end application QoS by: (1) specifying application QoS; (2) providing multiple configurations for the same application; (3) appropriate configuration selection; and, (4) QoS adaptation based on resource fluctuation, user mobility, and change in user preferences. However, no detailed description is given of the QoS parameters involved in the adaptation process, nor issues related to resource costs and user budget.

In their work on adaptive QoS control for multimedia wireless applications, the authors in [1] propose a middleware between the application and the underlying resource allocation and management functions of a generic wireless system. The main focus of their work is in the estimation of user preferences through a “user profiling” technique, where a user’s activity when accessing services is monitored to update his/her profile. The authors calculate relative media component importance based on the probability of a component being accessed and probability that it will not be degraded (both derived from previous user activity). In a 3D environment, however, we observe relative importance depending also on a user’s current viewpoint or position. Regarding calculation of resource allocation, the authors assume user quality of media components linearly depending on allocated bandwidth, which in many cases may not be the case (e.g. we show a calculation of allocated resources based on UFs which may be linear or nonlinear).

In [3], the authors propose an End-to-End Negotiation Protocol (E2ENP) supporting the active negotiation of QoS for multimedia. Service requirements are described using a hierarchical QoS specification (representing different abstraction levels). The authors assume an end-user application using this description model to derive valid QoS Contracts (service configurations) negotiated between users and enforced based on dynamic resource availability and user expectations. Although the authors consider user preferences in determining feasible *QoS Contracts*, no clear procedure is described for optimizing network resource allocation among service components. The authors focus on negotiation in a peer-to-peer architecture, whereas in our approach we address a client-server architecture with adaptation logic performed at a central network location. Such a design is influenced by 3GPP core network specifications where control functions are performed in the network.

An approach on handling issues of heterogeneity in NVR is given in [8]. A mathematical model is proposed to approximate the functioning of the NVR system considering individual user preferences and system constraints. Unlike our approach, the proposed system does not directly consider adaptation in response to changing network resource availability. Again, dynamic resource costs and user budget constraints are not directly addressed.

Specifically for NVR services, various approaches address the issue of changing requirements based on user navigation in a VE. The ISO/IEC 14772 Virtual Reality Modeling Language (VRML) standard, and its successor ISO/IEC 19775 Extensible 3D (X3D), include the notion of level of detail (LoD) where different quality levels are defined for objects and switched based viewpoint distance to the object. Applying area of interest (AOI) techniques leads to improved scalability [6].

While the mentioned approaches address various issues, none were found to combine all identified issues into a single model supporting dynamic QoS adaptation for complex multimedia such as NVR, as proposed in this work.

5. Conclusions and Future Work

The main contribution of the model proposed in this work is the procedure for end-to-end negotiation and dynamic adaptation of QoS parameters for NVR services, addressing the heterogeneous NGN environment. We compare our model to existing approaches, pointing out where we consider our work to extend or improve on exiting research. A generic client profile is introduced, while a proposed generic service profile serves to specify dynamic service requirements. Procedures for optimization of resource allocation and service configuration are outlined. While the model is applicable for any multimedia scenario, we choose to address the NVR service scenario as being one of the most ”complex” types of multimedia service. Model implementation shows how the use of dynamic QoS negotiation and adaptation mechanisms enables efficient use of QoS resources and achieves acceptable user perceived quality.

With regards to scalability issues, it is clear that running the optimization procedure for very many users when dynamic changes occur is definitely time consuming and costly. Besides the establishment of re-calculation thresholds, discrete solutions calculated in advance may be offered for particular combinations of constraints. Investigation of such issues is considered for future work. In addition, a key requirement is for dynamic service adaptation and QoS renegotiation to occur with minimal user perceived service disruptions. The amount of additional network traffic resulting from QoS related signaling is also a matter that needs to be considered, both in terms of cost and time.

References

[1] G. Araniti, P. De Meo, A. Iera and D. Ursino, “Adaptively Controlling the QoS of Multimedia Wireless Applications through ’User Profiling’ Techniques,” *IEEE J. Select. Areas Commun.*, vol. 21, no. 10, pp. 1546-1556, Dec. 2003.

[2] X. Gao, G. Wu and T. Miki, “End-to-end QoS Provisioning in Mobile Heterogeneous Networks,” *IEEE Wireless Comm. Mag.*, pp. 24-34, June 2004.

[3] T. Guenkova-Luy, A. J. Kassler and D. Mandato, “End-to-End Quality-of-Service Coordination for Mobile Multimedia Applications,” *IEEE J. Select. Areas Commun.*, vol. 22, no. 5, pp. 889-903, June 2004.

[4] M. Matijasevic, D. Gracanin, K.P. Valavanis and I. Lovrek, “A framework for multi-user distributed virtual environments,” *IEEE Tran. on Sys., Man and Cybernetics - Part B: Cybernetics*, vol. 32, no. 4, pp. 416-429, August 2002.

[5] K. Nahrstedt, D. Xu, D. Wichadakul and B. Li, “QoS-aware Middleware for Ubiquitous and Heterogeneous Environments,” *IEEE Comm. Mag.*, pp. 140-148, Nov. 2001.

[6] S. Singhal and M. Zyda, “Networked virtual environments: design and implementation,” *ACM Press SIGGRAPH Series*, ACM Press, New York, 1999.

[7] L. Skorin-Kapov, D. Mikic, D. Vilendecic and D. Huljenic, “Analysis of end-to-end QoS for networked virtual reality services in UMTS,” *IEEE Comm. Magazine*, vol. 42 no. 4, pp. 49-55, April 2004.

[8] H. Trefftz, I. Marsic and M. Zyda, “Handling heterogeneity in networked virtual environments,” *Presence*, vol. 12 no. 1, pp. 37-51, Feb. 2003.

[9] L. Skorin-Kapov, “Dynamic adaptation of QoS parameters for networked virtual reality,” MS. Thesis, University of Zagreb, FER, Zagreb, July 2004.

[10] X. Wang and H. Schulzrinne, “An Integrated Resource Negotiation, Pricing, and QoS adaptation framework for multimedia applications,” *IEEE J. Select. Areas Commun.*, vol. 18, no. 12, pp. 2514-2529, Dec. 2000.