Quality of Service Management for a Media-Enhanced Virtual Meeting Place

Evangelos Pappas-Katsiafas, Alan Smith, Kashaf Khan
Distributed Systems Group, BT Laboratories,
Martlesham Heath, Ipswich, IP5 3RE, England
email: [vang, asmith, kkhan] @jungle.bt.co.uk

Abstract: *This paper details work in progress to produce a Media-enhanced virtual meeting place using Quality of Service Management components to cope with heterogeneous networks and end-systems. In recent years there have been great advances in the fields of Virtual Reality, streamed Audio and Video, and network technology. The combination of these three technologies has the potential to produce a media-rich, realistic environment. Such an environment provides intuitive 3D navigation, interaction with other participants, and they selective audio-visual content. However, varying network performance, processing speed, and workload, have severe effects on the usability, performance and effectiveness of such an application. For this reason a Quality of Service (QoS) management is needed to adapt the application, and maintain usability and effectiveness at the highest level possible, under varying resource conditions.*

1 Introduction

The structure of this paper is as follows: Section 1 describes the background to this work. Section 2 describes our approach with the implementation details discussed in section 3. Section 4 discusses the Quality of Service (QoS) components that are used to adapt to different end-system and network capabilities and also to dynamically adapt to changes in resources and network usage. In section 5 we discuss some future work and we form some conclusions from our work in section 6.

The exponential growth of computer hardware performance combined with the latest advances in internet technology have opened new horizons to the development of collaborative virtual environments (CVEs.) In recent years we have seen a number of on-line virtual communities [14,15] growing significantly. Several CVE platforms and implementations have been proposed and developed so far, e.g. [12, 13]. Some of the virtual environments are enhanced, by incorporating video. Specific techniques for handling video in virtual environments have also been presented [16].

Virtual Reality: In the past few years advances in Virtual Environments have been significant. Although Virtual Reality (VR) is a relatively new area [7,8], today's computer hardware and software tools provide adequate platforms for virtual reality applications at reasonable cost. Some VR tools and technologies have become standardised (VRML[18], OpenGL[17], Java3D[19], Direct3D[20]) and even mainstream. Most of contemporary PC setups are capable of desktop VR, and other immersion devices are becoming cheaper and better. Game consoles are powerful and suitable machines for VR applications; the most recent versions of them feature an internet connection.

Internet evolution: The capacity and performance of the internet is constantly increasing. Both corporate and home connections are becoming faster and more reliable with both the development of old technologies i.e. analog modems, and the incorporation of new ones i.e. ISDN, cable modems, xDSL. Today, even home users can watch live video streams at acceptable quality, and play networked games over the internet. Video conference and internet phone applications are already used over the internet. However, the current internet protocols and infrastructure cannot guarantee QoS for these applications.

Multicast and new protocols: New protocols are being introduced for the reliable and guaranteed delivery of Real-Time media over the internet [9]. E.g. RTP/RTCP (RealTime Protocol, RealTime Control Protocol,) and RSVP (Reservation Protocol.) The description of these protocols is beyond the scope of this paper.

Another technology that is very useful in broadcast scenarios is *Multicast* [6] as opposed to *unicast*. Today's internet is based on unicast transmission, i.e. one sender sends one packet to one and only one recipient (1 to 1). Using multicast the transmitting node sends the data to a special multicast IP address. With IP Multicast there is an M to N communication between hosts.

Layered, streamed media: Streaming [1,2] allows the fast, and almost immediate delivery of multimedia content. However, streamed video and audio quality is very vulnerable to poor network performance, and the video or audio frequently gets interrupted if there is no additional mechanism to provide: i) guaranteed allocation of bandwidth, and/or ii) adaptation of the application to low bandwidth conditions.

Layered video and audio provide a solution for adaptation of multimedia content quality. The first layer contains the minimum part of the original signal, adequate to reconstruct a low quality version of it. Every successive layer increases the quality of the reproduced signal.

QoS: *Quality of Service* [3] management is needed for some applications to be effective and functional under varying host machine and network conditions. Large scale, computationally intensive applications, like networked Virtual Environments with multimedia content, fall naturally into this category.

The multimedia quality of the application can be adapted based on i) network traffic, and ii) host resources e.g. CPU power. For example, reducing the streaming media quality of the application when network traffic is detected, and graphic detail when host resources are low, will allow controlled degradation of quality without serious effect on the usability and the effectiveness of the application.

2 Our Approach

We are concerned with the design and development of an adaptable Virtual Environment for multimedia communication and collaboration among remotely placed participants. Our 3D Virtual Environment incorporates layered, multicast audio and video, spatial audio control, and 3D graphics technology. The combination of all the aforementioned features results in a realistic, easy to use and explore multimedia space.

Our CVE is built on top of our middleware platform Mware [5], which is responsible for the adjustment of QoS delivered in the virtual telecommunication space (see Section 4). The adaptation of the virtual space depends on i) network capacity, and ii) client machine capability.

Four components are adaptable:
a) the quality of audio, independently per audio source. Three levels (layers) of audio quality are offered.
b) the quality of video, also independently per video source. Four levels (layers) of video quality are available.
c) the level of movement detail of networked participants.
d) the quality of the 3D Virtual environment, e.g. renderable window size, shading method, texture presence and size, polygon complexity, avatar animation, etc..

2.1 Real-Time layered audio and video

The layered, multicast audio, and video is delivered by our *StreamGroups* library. This library is used to handle and configure the layered audio and video, and control the multicast transmission of them. Figure 1 exemplify the layered video encoding.

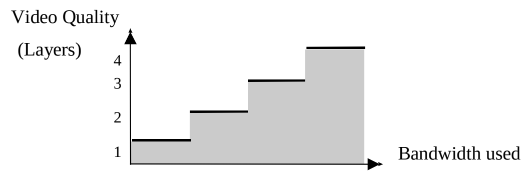

Figure 1 Layered video encoding

Video and audio signals are decomposed to layers during encoding. This way, we split the size of the original signal to a number of layers. The first layer, i.e. 'base' layer includes just enough information for a low quality reconstruction of the original signal. Its size is quite small compared to the original signal, thus it can be sent through a low bandwidth connection, e.g. 14 or 28 Kbits/s. In addition the decoding process of the 'base' layer is faster than the decoding process of the entire signal.
Consecutive layers add detail, hence improve quality, to the base layer. Every additional layer increase the size of information transmitted per time unit, hence higher bandwidth is needed. Moreover, higher decoding time is required. All layers together provide a high fidelity reconstruction of the original signal.
Layered audio and video provide a method of media quality adjustment. Applications can dynamically change the audio and video layers received, based on network performance and processing power. Therefore, applications using layered audio-visual components become more flexible and resilient.

2.2 Spatial QoS management

In virtual environments the user is usually represented by a human or other kind of figure, known as an 'avatar.' The quality of all video and audio sources in a Virtual Environment is adjusted based on the position and the orientation of the avatar representing the user. The closer a user gets to a video or audio source, the better the quality of the source, and the higher the audio volume. Audio and video quality control is achieved via the StreamGroups library described earlier in this section.
Figure 2 presents a plan of a virtual room with four video screens. It illustrates how Video quality varies based on the position and the orientation of the avatar:

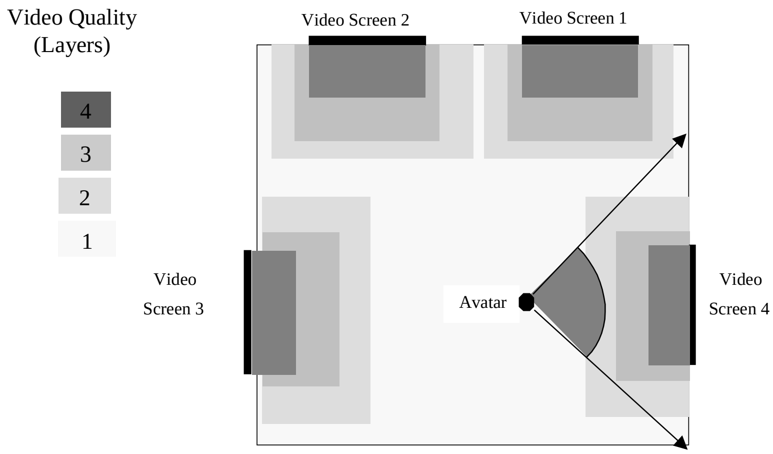

Figure 2 Room plan: Video quality zones

Each avatar has a specific field of view represented by the two arrow lines in Figure 2. In 3D graphics terms this is the 'view volume' of the avatar. Within the field of view, we choose an area of a certain distance in front of the avatar to be the area where quality will be increased. This area is represented by the dotted segment in Figure 2, and it's the '*area of interest*'. Media items out of the *'area of interest'*, appear in decreased quality, or they might not appear at all.
The quality of each media is determined by the intersection of the *'area of interest'* and the underlying 'quality zone.' For example, in Figure 2 the *'area of interest'* overlaps 'quality zone' 2 of the video screen 4. Therefore, video screen 4 appears in two layers, which is medium quality. Video content out of the field of view can be kept to the base layer, or it might be stopped completely.
The same spatial QoS mechanism applies to audio belonging to video screens. Audio has only three layers, hence three levels of quality. In addition audio volume varies according to the distance between the avatar and the video screen. Figure 3 presents two different methods of adjusting the audio volume based on distance.

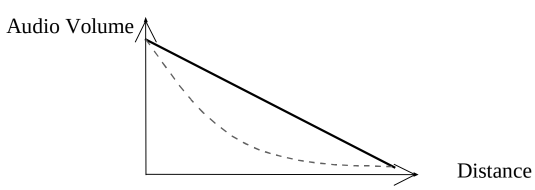

Figure 3 Audio adjustment methods

The black continuous line represents linear fading of audio volume, and the dashed line represents the function 1/x (where x is distance.)

All quality control requests are being passed to the QoS manager which decides about the level of quality of every participating media and the general quality of the application.

3 Implementation

There are currently two prototype applications demonstrating aspects of our approach discussed in section 2. Both of them present a virtual room with four video screens and their respective audio.

The first prototype demonstrates only spatial QoS. Therefore audio-visual quality and audio volume are controlled based only on avatar position and orientation.

The second is tightly integrated with the general QoS management architecture presented in section 4.3. This way, the quality of all the media components is handled by the general QoS manager, and is based in many aspects of machine and network heterogeneity.

The functionality of the application is exemplified by describing each figure appearing in appendix I.

Figure 4a*:* This picture presents a room with two participants and four video screens. Three streamed audio/video media are used along with a fourth local video source. It is a basic but expressive example. Participants can walk naturally in the room examining and listening to the content of the video screens. Three viewing modes are available for demonstration purposes:

a) Avatar independent, 'world view'. This viewing method is provided for demonstrating purposes. From this independent view, quality adaptations based on avatar movements are more obvious.
b) 'Avatar view'. This is exactly what the user would see if she was the avatar.
c) 'Camera following avatar' view.

Many video screens appear in the 'view volume' but the quality of them is not maximised, because they are not falling into the user's '*area of interest*.' In this particular view, video screens appear in small size and remote distance. It would be pointless, and a waste of resources to decode and render full quality.

The same principles apply to the audio quality. Audio quality is low because the avatar is not close to the audio source. For the same reason, audio volume is low too, as it naturally would be in the real world.

Figure 4b: This is an avatar view, taken in another room with a different arrangement of video screens. Another participant appears in the user's field of view. Designs of collaborative spaces can be different depending on the type of audience, and their purpose. For example, a corporate teleconference space will have different look and feel from a museum, or a children education room.

Figure 5: Many different participants appear in a room with four video screens and improved level of 3D quality by means of detailed avatars, shadows and reflections. Some more detailed avatars appear, like the avatar with grey suit, and the 'Al Capone' avatar wearing a hat. Also, this picture demonstrates real-time reflections and shadows of two avatars. Reflections and shadows improve significantly the level of reality, without being computationally expensive.

Figure 6: In this picture the spatial Quality of Service is demonstrated. Quality of video is adjusted by two factors: a) frame rate and b) compression ratio. Frame rate cannot be demonstrated by a still image, hence is not exemplified in this picture. The screenshot is taken from a remote point of view.

As the user-avatar approaches a video screen, the quality of the respective video and audio increases, e.g. maximum level for a user who concentrates exclusively on one screen. In this example, the avatar is located in front of the left screen. The quality of this screen is noticeably better. The face on the left screen appears more clear than the face on the right, although the face of the actor on the left screen is smaller than the face appearing on the right screen. Also the right screen has more 'jagged' edges and blocks of solid color due to the high compression ratio. In addition, the user listens to the audio of the left screen only, in full quality and volume.

Figures 7a, 7b: The same concept of Spatial adaptation of quality is demonstrated in these two pictures. The video quality of the screen, which the avatar is close to, is at the maximum level possible. The video screen on the right, which is out of the avatar view, appears with much lower quality. The images on the left screen appear much clearer and 'crispy' than the fuzzy ones on the right screen.

The current prototype has been implemented on Windows NT 4.0. However most of the components of our architecture are platform independent. Layered, streamed audio and video components are a BT implementation of the H263+ standard for video and audio streaming. The middleware platform is the BT Mware [5] platform. For the 3D virtual world we use OpenGL 1.1, and the GLU, and GLUT 3.7 [22] utility and windowing libraries respectively.

4 General Quality of Service Management

4.1 QoS Adaptation

The role of QoS management is to adapt to heterogeneous end-systems and networks so that users can participate in a session regardless of what the power of computer or network bandwidth they are using. It is also adapts to changing requirements in the virtual meeting place, e.g. adapting to an avatar's position in the virtual world and changing the quality of video, say, as an avatar draws nearer to a screen.

We are using an internet environment for our network so we have M-N multicast. We can use the same multicast address to send video and audio from different sources. This has some implications for our QoS management. The adaptation techniques we are using can be used to adapt to end-system resource and/or network resource availability. The layered media codecs used for audio and video can be used to adapt to just end-system resource, or to both end-system and network resources, depending on the prevailing conditions. We also introduce the notion of layered data handling which can be used to adapt in a similar fashion. The level of 3D graphics can be adapted purely to control the amount of end-system resource that is used.

4.2 QoS Management Architecture

This is shown in Figure 8. It is important to note we have 2 levels of QoS manager:

- Individual QoS manager - monitor and adapt the performance of a particular media component
- General QoS manager - arbitrates between individual QoS managers

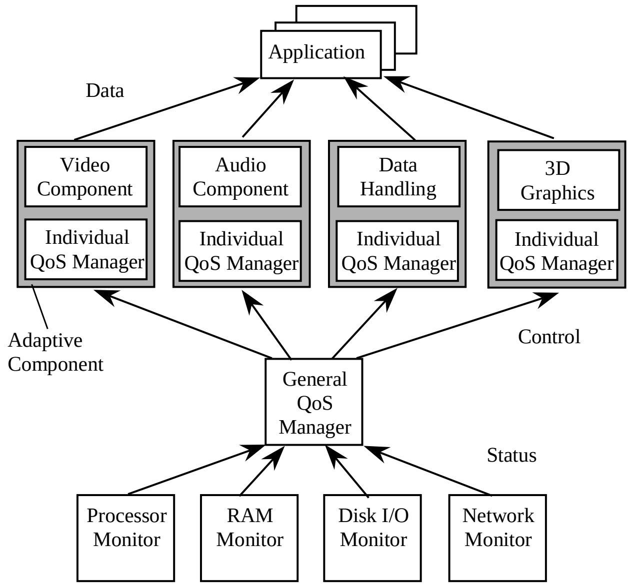

Figure 8 QoS management design

Individual QoS managers interface to media components (video, audio, data handling and 3D graphics) that are able to adapt to changing resources. We refer to this combination as an adaptive component. The individual QoS managers control how and when adaptation takes place in consultation with the general QoS manager. Also, they are responsible for monitoring the performance of the media components and adapting their performance. The general QoS manager is responsible for monitoring the system state and for arbitrating where to use resources.

The media components pass status information to the individual QoS managers, for example:

- Failure to decode a received frame within a specified time span
- The percentage of lost packets it has detected.

Given this information the individual QoS manager is able to detect if the media component is starved of resources and will inform the QoS manager which must then decide which component must decrease its resource usage.

The general QoS manager's role is one of arbitration and monitoring to ensure that resources are directed to the components with the highest priority. This priority is determined by the collaborative environment application.

The general QoS manager receives reports on:

- system and network performance from the monitors and
- status reports from the individual QoS managers as to how the media components are performing.

When the monitors detect spare resources (a low water-mark threshold), the general QoS manager sends an " increase resource usage" message to the component with the highest priority that is not at its maximum level. In the case of failure reports from individual QoS managers the general QoS manager chooses the component with the lowest priority that is not at its lowest level, and sends it a "reduce resource usage" message. Each reduce or increase message will have a “reason” parameter. The reason will be either network, end-system, or both. The individual QoS manager will adapt accordingly.

4.3 Adapting to Host Resource

We can adapt to host resource on a per source basis. That is to say, we can adapt the quality of stream received from each source in order to use more or less computing power on the user’s end-system. The collaborative environment application passes a list of priorities for all the components and lists of priorities for sources within those components. The QoS manager will attempt to fulfil maximum QoS on the component with the highest priority. When an audio or video individual QoS manager receives a reduce resource usage message with a reason parameter of “end-system” it will attempt to use less resource by reducing the number of layers being decoded for the source with the lowest priority. When an individual QoS manager receives an increase resource usage message it will increase the number of layers being decoded for the source with the highest priority not at the highest level.

4.4 Adapting to Network Resource

When network congestion is detected we must drop a layer. Thus the video quality will decrease for all TV screens no matter what the priority is. When there is no congestion detected we can try adding the layer back. The layer being decoded will remain one level down for all TV screens until the end-system resource monitors have been checked to ensure spare resource is available.

Future research will include looking at how we can group sources together so we can adapt better for network resource availability. We currently have every source multicasting to one multicast address. At the other extreme we could have one source multicasting on one address. We could form groups of sources, so each group uses the same multicast address or addresses. This is efficient when sources have fixed position, or sources move slowly and form groups.

There is a trade-off between individual end-system adaptability and the number of multicast trees required to support a session. From the end-system point of view the most beneficial scheme would be to have one multicast group per sender. In terms of re-use of multicast routing it would be best to have one multicast tree.

4.5 QoS Controlled Data Handling

Participants in a Collaborative Virtual Environment are continuously emitting various type of data into their environment, e.g. movement, facial expressions, text, audio, video etc. This communication can occur in the form of QoS controlled data handling, which is particularly useful in a heavily populated, busy virtual world. We are using several mechanisms to allow adaptation. These can be classified into two types:

- Protocol adaptation
- Data manipulation

Protocol adaptation includes manipulation of the stack as has been previously demonstrated in Ensemble [4] and Horus. An application may start by using a protocol initially selected for high reliability, but due to significant congestion on certain routes a more forgiving protocol may be used instead.

Data manipulation comprises compression and layering of discrete data transmissions. Compression can help overcome the lack of network resource, but at the expense of increased resource usage on the end-system.
Layering requires the applications to prioritise their data. In the virtual world example movement data would be assigned a lower priority than the final resting position of an avatar. An end-system can then leave the layer containing the lower priority data, but the user can still participate in the virtual world to some degree. Figure 9 illustrates the principle behind this.

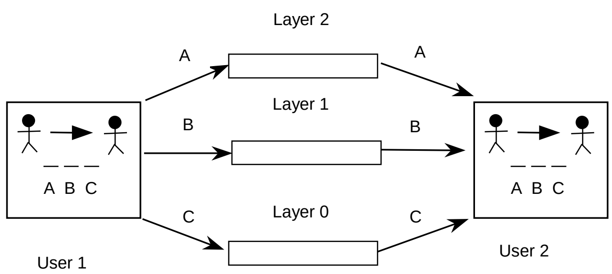

Figure 9 Layered Data Transmission

4.6 QoS Controlled 3D Graphics

'*3D Graphics'* QoS manager monitors: i) processor performance, ii) presence and performance of dedicated hardware acceleration, iii) video and system memory.
Based on the above measurements, QoS manager adjusts the following 3D features of the application: a) renderable window size, b) polygon complexity, c) shading and lighting method, d)presence and size of textures, e) avatar animation quality, f) real-time realistic effects, i.e. shadows and reflections, g) presence and complexity of decorating items.
The prioritisation of the above list depends on the QoS manager policy. Many levels of 3D quality can be formed. A simplified 3D quality adaptation is presented in [10.] A 3D graphics and video quality control method is presented in [11.]

5 Future Work

The next evolution of the Mware based virtual environment will involve *positional collaboration.* This way, besides audio and video data, positional data transmitted by each participant will be rendered at every client.
Extensions of the *QoS control mechanism* to adapt to new services/media. Hence, new services may be dynamically adapted depending on the host and network resources availability. This mechanism could be used for example to adapt, add, remove adjustable 3D graphic features as listed in section 4.6.
Synchronisation of Datastream. This involves extending the RTP time-stamping mechanism so that other services which are also streaming data may be time-stamped with the same time base. This could be used to implement a real-time position and orientation service for participants.

3D Spatial Audio. This service will enhance the level of reality in the environment by enabling the participant to distinguish individual sound sources, which will seem to be emitting from a particular direction and position.

6 Conclusions

We tackled the issue of QoS management in a collaborative video-enhanced virtual environment by integrating a QoS manager mechanism, which is part of our middleware platform Mware. Consequently, the virtual environment becomes more flexible, adaptable, and capable of performing better under varying conditions of network traffic and host resources. This is a key benefit for the computationally intensive, VR applications whose performance degrades significantly, and usually in an uncontrolled manner, when host and network resources decrease.

We aim to conclude our current implementation, and expand our design towards increased audio/visual quality, multi-group management, QoS policy, and other aspects discussed in the future work section.

7 References

[1]: 'Receiver-driven Layered Multicast', McCanne S, Jacobson V, and Vetterli M., ACM SIGCOMM, August 1996, Stanford, CA, pp. 117-130.

[2]: 'Layered Audio-visual Coding for Multicast Distribution on IP Networks', Nilsson M, Dalby D and O'Donnell J., in preparation.

[3]: 'Quality of Service Management within a Middleware for Multicast Applications ', Smith A, Jacobs R., Workshop on QoS Support for Real-Time Internet Applications, June 1999.

[4]: M Hayden. The Ensemble System, PhD Dissertation, Dept. of Computer Science, Cornell University, USA, 1997.

[5]: http://www.labs.bt.com/projects/mware

[6]: 'IP Multicasting', Dave Kosiur, John Wiley & Sons Inc. 1998.

[7]: ACM Transactions on CHI, Special issue on Virtual Reality Software and Technology, Vol.2, No 3, September 1995.

[8]: 'Computer Graphics: Principles and Practice' 2nd Edition. Foley, VanDam, Feiner, Hughes, Philips, Addison-Wesley publications.

[9]: 'Internet QoS: A Big Picture', Xipeng Xiao and Lionel M. Ni, IEEE Network, March/April 1999, Vol 13, No 2.

[10]: 'Soft Real-time Application execution with Dynamic Quality of Service Assurance', IFIP, IEEE, IW QoS 1998.

[11]: 'Adynamic Qos Adaptation Mechanism for Networked Virtual Reality', IFIP, IEEE, IW QoS 1997.

[12]: "Networked Virtual Environments : Design and Implementation", Sandeep Singhal, Michael Zyda, Addison-Wesley Pub Co; ISBN: 0201325578.

[13]: Supporting Cooperative Work in Virtual Environments', The Computer journal, Oxford press, 1995.

[14]:http://www.activeworlds.com

[15]:http://www.blacksun.com

[16]: 'Techniques for handling Video in Virtual Environments', Gianpaolo Carraro, John Edmark, J. Ensor, 25th Annual conference on Computer Graphics, 1998.

[17]: http://www.opengl.org

[18]: http://www.vrml.org

[19]: http://www.sun.com/desktop/java3D

[20]: http://www.microsoft.com/directx/overview/d3d/default.asp