Application Level QoS Modeling for A Distributed Multimedia System

Kazutoshi FUJIKAWA†, Shinji SHIMOJO‡, Hisashi SHIMAMURA‡,
Yuuichi TERANISHI‡, Toshio MATSUURA‡, Hideo MIYAHARA‡

†Graduate School of Information Science,
Nara Institute Science and Technology,
Ikoma, Nara 630-01, Japan

‡Department of Information and Computer Sciences,
Faculty of Engineering Science, Osaka University,
Toyonaka, Osaka 560, Japan

Abstract

When current multimedia applications could not keep a certain level of quality by the lack of CPU power and/or network bandwidth, they can not achieve their requirement and the intention of applications may be lost. Therefore, the mechanism to preserve the intention and specify its requirement is needed.

In this paper, taking a distributed multimedia presentation system as an example, we propose a new definition of application level QoS which includes quality of synchronization and integration of multiple media as well as quality of a single medium. We also define a few measure for specifying QoS and its priority used when renegotiation or re-schedule is required. We define a set of QoS parameters as a policy which specifies an intention of a presentation. We propose a system architecture which initiates QoS negotiation among servers, a client, and a transport system and re-negotiates and re-schedules QoS.

Keywords

distributed multimedia system, QoS, adaptation

1 Introduction

High speed networks and multimedia workstations or personal computers are enablers of distributed multimedia applications such as desktop video conferencing or multimedia information services. *WWW(World Wide Web)*[5] and *gopher*[6] are actual examples of a multimedia information services in wide area network. Such kind of applications as WWW or gopher can become sophisticated to a distributed multimedia information services where a client retrieves multimedia data from different servers on the fly and plays back at client's site in an integrated fashion[3, 4] when connecting with high speed wide area networks.

Since these applications run on various system environment such as network, server/client machines, and operating system, the QoS (Quality of Service) should be guaranteed. Some type of transport systems (we call a system which support below the transport layer a *transport system*) such as ATM, ST-II[8], and HeiTP[7] in data link, network, and transport layer protocols can guarantee QoS, while other type can offer only best-effort QoS like TCP/IP. Since QoS parameters provided by the transport system are rather simple and concrete such as delay, jitter, latency, and bandwidth, they don't match the requirement of an application straightly. Therefore, application level QoS parameters should be defined and mapped into lower level QoS parameters properly. Especially, when considering QoS in application level, QoS for multiple media such as video, sound, or animation should be integrated and synchronization of their streams is also considered.

Before starting an application, negotiations among servers, a client, and a transport systems should be performed. While executing the application, renegotiation or re-scheduling are also required due to changes of the load in network and CPUs of server and client.

In this paper, taking a distributed multimedia presentation system as an example, we propose a new definition of application level QoS which includes quality of synchronization and integration of multiple media as well as quality of a single medium. We also define a few measure for specifying QoS and its priority used when renegotiation or re-schedule is required. We define a set of QoS parameters as a policy which specifies an intention of a presentation. We propose a system architecture which initiates QoS negotiation among servers, a client, and a transport system and re-negotiates and re-schedules QoS.

In the next section, we describe the model of distributed multimedia system. In section 3 and 4, we discuss transport level QoS and application level QoS respectively. In section

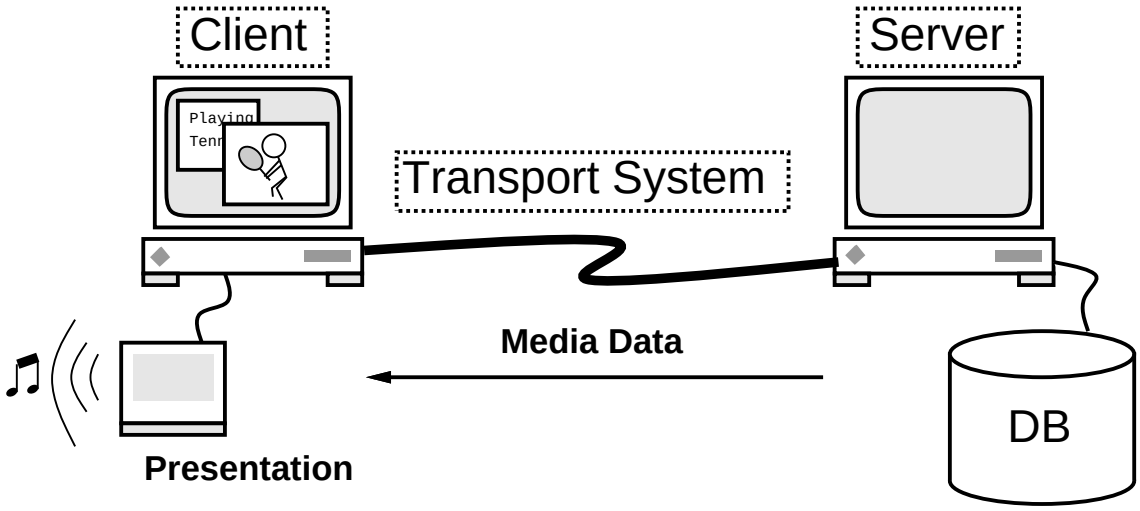

Figure 1: Model of a multimedia presentation system

5, we mention the method of adjustment of QoS. In section 6 and 7, we describe the architecture of multimedia information system with QoS control mechanism and examples of QoS control. We discuss the related works in section 8 and conclude in section 9.

2 Model of distributed multimedia information services

First, we define a model of distributed multimedia information services. Distributed multimedia information services consists of multimedia servers and a client connected with a transport system (Figure 1). The client retrieves multiple multimedia data on the fly from servers along with a scenario or a description while each server sends the data to the client via the transport system. Multimedia data should be arrived and presented in integrated fashion satisfying *application level QoS* described in the scenario or the description.

To achieve application level QoS, the client analyzes its scenario and translates it into *transport level QoS* considering capacity of CPU, memory, network bandwidth, and so on from given *system environment profile*. At the same time, the client chooses appropriate servers and negotiates with them for appropriate seize or resolution of requiring multimedia data to suit for system environment profile.

Some type of transport systems, ATM, ST-II[8] and HeiTP[7] in data link, network and transport layer protocols can guarantee QoS, while other type can offer only best-effort

Table 1: QoS Parameter in the Transport System

QoS parameter	Meaning
delay	time spend in data transmission
jitter	scattering among receiving data
throughput	bandwidth
error rate	probability of error occurring

QoS such as TCP/IP. In case of guaranteed transport system, the client requests a desired transport QoS to the transport system and initiates a session, while in case of best-effort transport system nothing can be performed at that time. When the application starts, media data are transfered from the servers along with the schedule and displayed at the client. Due to the CPU or network overload and underestimation of QoS parameters, multimedia information may not be transmitted or played in full quality. In this case, the client decides to degrade the original QoS, that is, *adjusting QoS* and initiates renegotiation and re-scheduling with the servers and a transport system.

3 Transport level QoS

There are many discussion about QoS below the transport layer, but there is no widely accepted protocols right now. Therefore, we adopt the common features among these protocols and define QoS in a transport system.

In ATM and ST-II, we can specify the QoS parameters shown in Table 1 and they can guarantee specified QoS parameters[8]. In ATM, several service classes are provided for a set of QoS parameters. The desired QoS is achieved by selecting a corresponding service class.

Best-effort type transport system such as TCP/IP can not guarantee QoS in contrast to the ATM and ST-II. However, even in ATM and ST-II, if requested QoS can not be achieved because of congestion of the network, the connection request between server and client at desired QoS is not admitted and connection establishment is postponed or rejected.

4 Application level QoS

In this section, we describe the concept of the QoS in a distributed multimedia information services and its adjustment. These QoS parameters should be specified in a scenario or a description as a policy.

In the application level QoS parameters, there are two types: *intra-media QoS parameters* and *inter-media QoS parameters*. Intra-media QoS parameters relate to only a single medium, while inter-media QoS parameters relate to the relationship among several media.

As intra-media QoS parameter, we consider a temporal scale such as frame rate of motion video and a spatial scale such as resolution of image. Intra-media QoS parameters are specified in the form of the minimum request of quality of a single medium and the priority among several QoS parameters within the medium. As inter-media QoS parameter, we consider synchronization and its priority among several media. Inter-media QoS parameters are specified in the form of the accuracy of synchronization and the priority among several media.

4.1 Policy of a scene

Since a scene in a scenario consists of multiple media and their combination, the scene has a certain set of QoS requirements for media and their combination. When considering renegotiation and re-scheduling, it has a certain way to degrade the original QoS parameters. We call these set of requirements in a scene a *policy*. A policy is described in a scenario and differs from a scene to a scene.

By introducing a notion of policy, we can have templates of a policy. We provide several typical templates representing specific policy and a user only selects one of them and customizes it.

4.2 Intra-media QoS parameters

Intra-media QoS parameters designate how a single medium is displayed and which parameter has significance when renegotiation or re-scheduling are occured. We specify these by four parameters and their preferences.

QoS parameters

In QoS parameters, we specify the maximum requirements of parameters where these quality is preferred to in a ideal situation. We have following four parameters as follows:

Latency

A time required for media data coming from a server.

Response

A time required for a presentation to begin at first time.

Crops

Total rate of skipped information.

Continuity

The number of sequence of processed information.

Since these parameters are generalized, they can be transformed into a specific measures suited for each medium. For example, *crops* parameter for motion video can be considered as a rate of number of actually displayed frames and ideal number of frames is 30 in a second.

Property preference

Property preference specifies the priority among the above parameters within a single medium.

4.3 Inter-media QoS parameters

Inter-media QoS parameters designate how multiple media played simultaneously should be displayed and which medium has significance. We specify these by three *synchronization types* and *media preference*.

Synchronization types

Three synchronization types(Figure 2) are defined to designate how different media displayed simultaneously synchronized as follows:

- Realtime

  Each medium in a scene is displayed in real time fashion. Some contents may be lost.

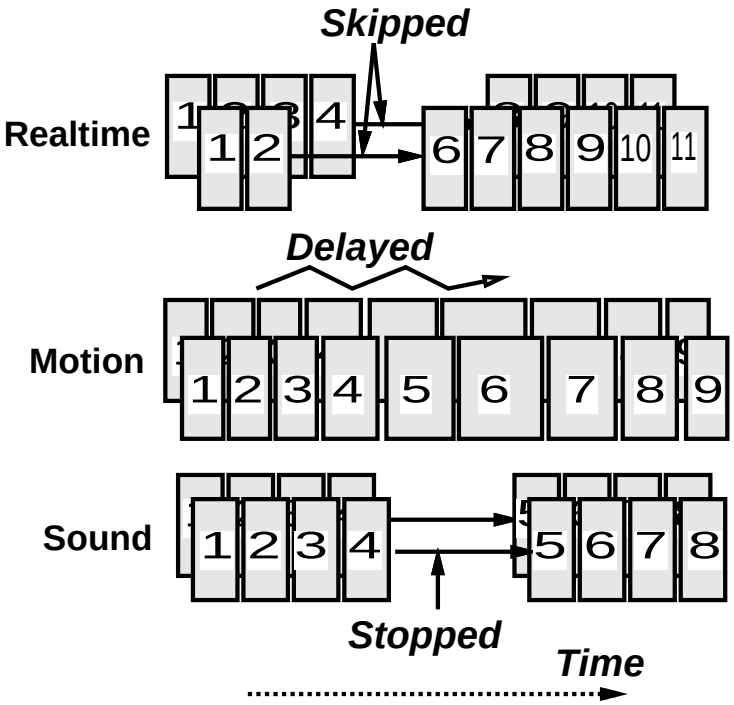

Figure 2: Synchronization types

- Motion

  Each medium is displayed without losing any contents by changing the processing rate. Each medium is not paused.

- Sound

  Each medium is displayed without losing any contents. Each medium is paused if the information to be displayed is not prepared.

Media preference

Media preference specifies the priority among several media data which will be played simultaneously. The medium with lower priority is degraded at first when renegotiation or re-scheduling is occured.

5 Adjustment of QoS

To achieve QoS in application level, two adjustments are required, one is before the execution of an application and the other is in the middle of the execution of the application. QoS parameters in application level are translated into transport level QoS while negotiating servers and transport system before the application starts. If the requested QoS parameters to transport system can not be satisfied, the QoS parameters should be modified and renegotiation and re-scheduling will be occured.

Moreover, the transport system can not preserve the QoS certainly. If the transport system can not preserve the QoS, media scaling, buffering media data by a application, or staggering time axis of a presentation may be performed to prevent the quality of media data from deterioration.

Now, we show the examples of adjustment of the QoS during the execution of an application. If the transmission of motion video is delayed, we can find two countermeasures. One is to keep displaying the contents of motion video in real time by degrading temporal scaling or spatial scaling such as skipping the frames or lowering the color quality. Another is to display all frames certainly by extending the time axis. Moreover, in case of a presentation including sound data, we may wait for the arrival of the whole sound data by pausing the play of other data synchronized with the sound data and play the sound data in order to recognize the contents by keeping the pace of playing. In opposite, we may display the data immediately by degrading the quality of media.

5.1 The System environment profile

For the adjustment of QoS, *system environment profile* is required to consult system and network performance. The *system environment profile* is a represents static information such as the capacity of CPU and dynamic information such as the load of CPU or network. The system environment profile includes information about server, client, and network as following:

- server information
  - static information
    - data format of media which the server can process
    - capacity of the CPU

  - dynamic information
    - load of the CPU
    - utilization of buffer for transmitting data
- client information
  - static information
    - data format of media which the client can process
    - capacity of the CPU
  - dynamic information
    - load of the CPU
    - utilization of buffer for receiving data
- network information
  - static information
    - bandwidth between server and client
  - dynamic information
    - congestion of network

The static information is used to estimate the suitable quality of a presentation before the presentation is executed. The dynamic information is used as required during the execution of a presentation.

5.2 Substitute and bridge

When the requested QoS can not be satisfied, another media data or a scenario may be transmitted and displayed instead. We propose two mechanisms to compensate this situation, *substitute* and *bridge*.

The examples are following:

- substitute

  If sound data can not be transmitted preserving the minimum quality, a text information should be transmitted and displayed instead of the sound data. If motion video data can not be transmitted preserving the minimum quality, another motion video with the same contents and the less resolution may be transmitted.

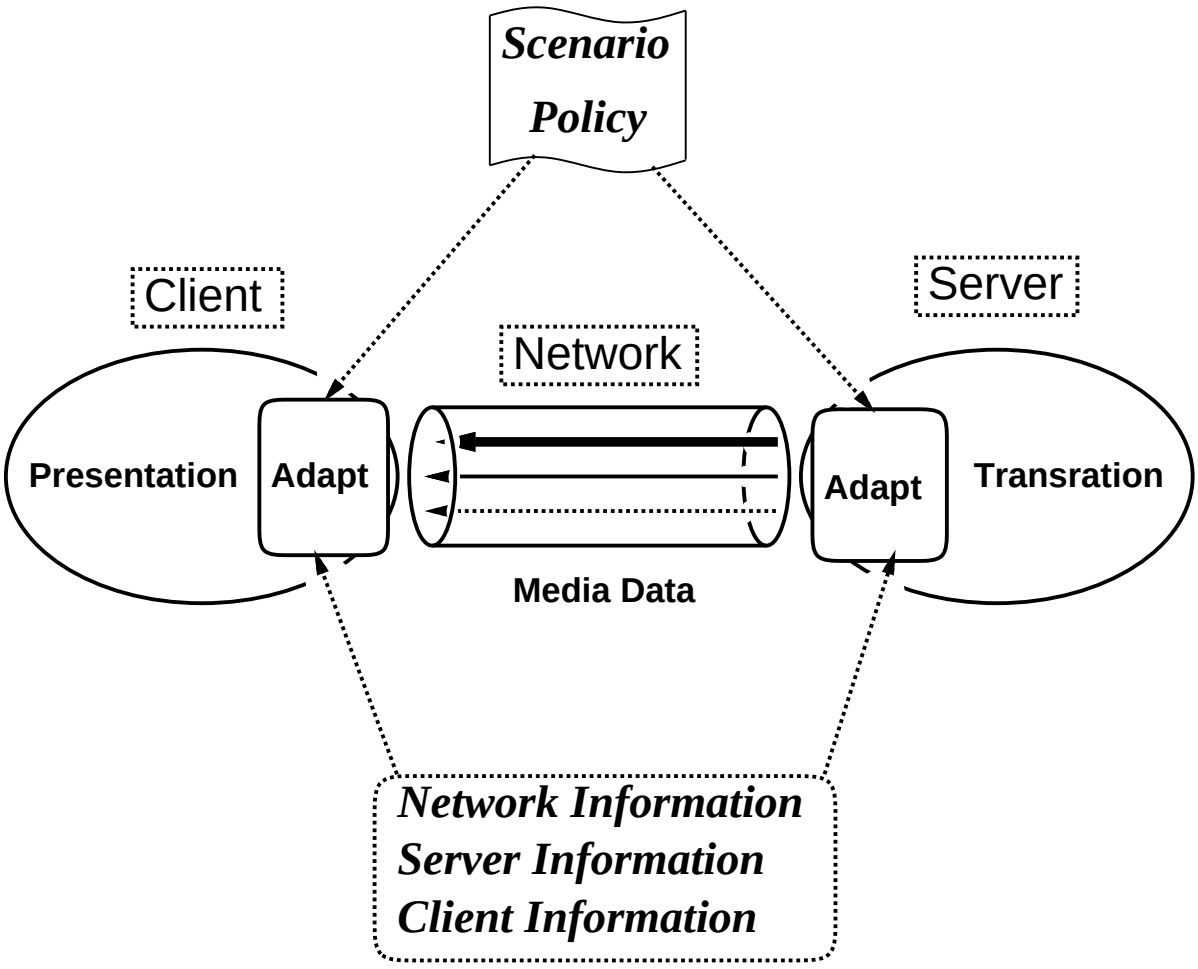

Figure 3: Model of a multimedia presentation system with QoS control mechanism

- bridge

  If the preparation time to play back data is needed too much and the data can not be played back immediately, another data prepared beforehand will be displayed.

6 Architecture model of a distributed multimedia system

In this section, we describe the architectural model(see Figure 3) of distributed multimedia system with the capability of *environmental adaptation*. We call the adjustment of QoS based on system environment profile environmental adaptation.

The environment adaptation is performed in both server and client. The server carries out environment adaptation related to data transmission. First, the server calculates

client's processing rate for playing back media data based on its MIPS value of static client information. And then, the server estimates client's actual processing rate from its load average or the number of processes running of dynamic client information. Furthermore, the server determines its actual transmission rate by comparing the transmission rate estimated from the bandwidth of network of static information and round trip time between server and client, and client's estimated processing rate.

Next, the server interprets the scenario of a presentation and retrieve the media data described in the scenario. And then, the server performs environment adaptation based on the policy of the scenario in order to transmit media data corresponding to the above transmission rate. If the QoS parameters in the policy is not satisfied, the server selects the media data of lower priority referring the policy parameters about media preference. The quality of the selected media data is deteriorated referring the policy parameters about property preference. Moreover, the transmission rate may be modified according to change of the status of client and network, and the quality of media data is adjusted.

The client puts the transmitted data into its buffer. The buffer is provided in order to cope with fluctuation of the load of client. If the buffer is overflowed, it represents that client's actual processing rate is not sufficient. In this case, client performs environment adaptation based on the policy. Mainly, the environment adaptation is to skipping the certain contents of data. At the same time, the client requests the server to decrease the transmission rate.

If the buffer is underflowed, it represents that the processing power of client is enough. In this case, client does not perform environment adaptation. At the same time, client observes the status of the network. If the bandwidth of the network is enough, client notifies server of ability of processing more data. Otherwise, client does nothing.

7 Example of environment adaptation

In this section, we show the example of environment adaptation. We consider tennis lesson as an example and explain how to adjust the limitation of system environment. The scenario of tennis lesson consists of swing coaching scene and interview scene. In each scene, the presentation is executed using motion video, animation, sound, and text.

Swing coaching

In this scene, the instruction of tennis swing is presented using motion video of swing and explanation sound and text. Since showing the motion of the swing exactly is very important, the priority of motion video is specified higher than other media in the policy. In this scene, as the explanation text is displayed, the priority of the sound of the same explanation is specified lower. Moreover, as the motion of the swing is important within motion video media, the frame rate is given higher priority than color scale and resolution. The policy of this example is the following:

- QoS parameters
  latency $\cdots$ 3, response $\cdots$ 3, crops $\cdots$ 5, continuity $\cdots$ 5
- Synchronization type
  motion
- Media Preference
  sound $\cdots$ 2, motion video $\cdots$ 5, text $\cdots$ 5
- Property Preference
  resolution $\cdots$ 3, color scale $\cdots$ 2, display size $\cdots$ 4, frame $\cdots$ 5

Based on the above policy, when it becomes impossible to process all data with full quality, the quality of the sound data is deteriorated first. If it is still insufficient to process data, the quality of the color scale which is lower priority within the motion video is deteriorated. Thus, the quality of a scenario is deteriorated in order of lower priority. The ratio of quality deterioration is determined from the value of the priority and the characteristic of media.

Interview

In this scene, questions and answers to the instructor is carried out. The media used in this scene are sound and motion video. Since the sound of questions and answers is more important than the motion video, the priority of the sound is put higher in the policy of this scene. The policy of this scene is the following:

- QoS parameters
  latency $\cdots$ 3, response $\cdots$ 3, crops $\cdots$ 5, continuity $\cdots$ 5

- Synchronization type

  sound

- Media Preference

  sound $\cdots$ 5, motion video $\cdots$ 4

- Property Preference

  resolution $\cdots$ 5, color scale $\cdots$ 4, display size $\cdots$ 4, frame $\cdots$ 3

In this policy, when it is impossible to process all data in real time, responsibility is made a sacrifice of and the presentation is suspended. At this time, a certain amount of data is transmitted to the client before the presentation is resumed. The quality of the sound data is not deteriorated, but the certain frames of the motion video data is skipped. Also, in this case, quality deterioration is performed in order of lower priority.

8 Related Works

Heidelberg Transport System[7] provides *media scaling* to keep the consistency of the presentation by degrading the quality of multimedia information when the requested QoS can not be archived in the transport system. In media scaling, in order to estimate the QoS of the presentation, the minimum request of QoS parameters should be specified. However, they treat media scaling with in a single media.

The idea of coordinating multiple media in distributed multimedia system is also proposed in [9] where an orchestrator which take A/V elements and match them to the A/V servers with appropriate responsiveness. An orchestrator monitors network or host performance and coordinate A/V elements. Although they show a general implementation of an orchestrator, CircusTalk, they did not show specific implementation of application level QoS. Campbell, et. al proposes an architecture of QoS across all layers[10] and its implementation on distributed real-time environment with micro-kernel operating system, Chorus[11].

9 Conclusion

In this paper, we proposed modeling of a distributed multimedia system with QoS control mechanism for the system environment. There are no multimedia information system which

can perform proper quality control. In order to cope with this problem, we introduced the concept of the policy which can represent the intention of a presentation as the priority of the quality of media data. Referring to the policy, it is possible for a multimedia information system to adjust QoS reasonably without violating the consistency of a presentation.

The problem to be solved is to establish the detail algorithm of QoS control using the system environment and the policy. Especially, the algorithm must reflect the priority of the quality deterioration. Furthermore, it is necessary to investigate the method for detection of the load of the processor and the network. Finally, we have to implement an actual multimedia information system with such QoS control mechanism.

References

[1] Shimojo,S., Matsuura,T., Fujikawa,K., Nishio,S., and Miyahara,H : *"A New Hyper-object System Harmony: Its Design and Implementation"*, International Conference on Multimedia Information Systems, McGraw-Hill, pp.243-257, Jan. 1991.

[2] Fujikawa,K., Shimojo,S., Matsuura,T., Nishio,S., and Miyahara,H : *"The Synchronization Mechanisms of Multimedia Information in the Distributed Hypermedia System Harmony"*, Proceedings of the 1st International Conference on Multi-Media Modeling, pp.275-289, Singapore, November, 1993.

[3] Bulterman,D.C.A. : *"Specification and Support of Adaptable Networked Multimedia"*, ACM/Springer Verlag Multimedia Systems Journal, Vol.1, No.2, pp.68-76, September, 1993.

[4] Blakowski,G., Hubel,J., and Langrehr,U. : *"Tool Support for the Synchronization and Presentation of Distributed Multimedia"*, Computer Communications Vol.15, No.10, pp.611-618, December, 1992.

[5] Berners-Lee,T., et.al. : *"World-Wide Web: The Information Universe"*, Electronic Networking: Research, Applications, and Policy, Vol.1 No.2, Spring, 1992.

[6] McCahill,M. : *"The Internet Gopher Protocol: A Distributed Server Information System"*, Connexious – The Interoperability Report, Vol.6, No.7, July, 1992.

[7] Delgrossi,L., et.al. : *"Media Scaling for Audiovisual Communication with the Heidelberg Transport System"*, ACM Multimedia '93, pp.99-112, August, 1993.

[8] Topolcic,C. : *"Experimental Internet Stream Protocol, Version 2 (ST-II)"*, RFC1190, October, 1990.

[9] Gutfreund,Y., et.al.: *"CircusTalk: An Orchestration Service for Distributed Multimedia"*, ACM Multimedia 93, pp.351-358, 1993.

[10] Campbell,A., et.al.: *"A Quality of Service Architecture"*, Computer Communication Review, Vol. 24, No. 2, Apr. 1994.

[11] Robin,P., et.al.: *"Implementing a QoS Controlled ATM based Communications System in Chorus"*, Internal Report No. MPG-94-05 Department of Computing, Lancaster University, March 1994.