Application Design for Cooperative QoS Management

*Stefan Fischer, Mohamed-Vall O. M. Salem and Gregor v. Bochmann
University of Montreal, Department IRO*

*C.P. 6128, Succ. Centre-Ville, Montréal (PQ) H3C 3J7, Canada,
Phone: +1 514 343 6111 ext. 3504, Fax: +1 514 343 5834, Email:
{fischer,salem,bochmann}@iro.umontreal.ca*

This work was supported by a grant from the Canadian Institute for Telecommunications Research (CITR), under the Network of Centers of Excellence Program of the Canadian Government.

Abstract

We developed a new QoS management scheme called Cooperative QoS management which is especially suitable for multicast multimedia communication. In this paper, we discuss how this new scheme influences the design of applications based on it.

Keywords

QoS management, MM application design, multicast

1 INTRODUCTION

In multimedia applications including multicasting to many users, such as teleconferencing or educational applications, the QoS management approach where QoS levels are negotiated between sender and receiver is not workable anymore, because the number of users involved is too large. For instance, negotiation of QoS parameters between the sender and every single receiver becomes impossible, since (1) the system would quickly become overloaded and (2) it would have to take into account (and possibly provide) many different qualities requested by users. Instead, a more decentralized approach seems suitable, where QoS management functions such as QoS negotiation, adaptation or renegotiation are distributed over the network. We developed such an approach called Cooperative QoS Management (Fischer, Hafid, v. Bochmann and de Meer 1997), where so-called QoS agents are installed on the routers and end systems participating in an applications.

After a brief introduction into the principles of Cooperative QoS Management in Section 2, this paper gives, in Section 3, some hints how the design of applications is affected when based on this new scheme, using a teleteaching

application as an example. An extended version of the paper is available as technical report from the authors.

2 COOPERATIVE QOS MANAGEMENT

Cooperative QoS Management has been developed with multimedia applications in mind, in which many users participate at the same time, such as teleeducation systems or live video transmissions of major sports events. We assume that single data streams are multicast to many users and that senders offer the same media stream in several qualities, e.g. a high, a medium and a low quality video stream. There are no individual QoS negotiations between senders and receivers; rather, receivers have to select among the qualities offered by the senders. However, QoS negotiations are carried out between receivers and *QoS agents* which are installed throughout the system. This makes up a major difference of our approach to a similar one discussed for the MBone (McCanne, Jacobson and Vetterli 1996), where no negotiations take place at all.

The QoS agents are able to communicate with their neighboring agents, informing them e.g. about current QoS values supported in their local area or about possible QoS problems. This knowledge is basically application-oriented, i.e. the agents know about QoS requirements and negotiated values for users as well as relationships between streams. This constitutes a main difference of our approach compared to existing QoS management functions on network nodes which deal with lower-layer QoS, such as ATM cell loss priority etc., and which do not have any information about relations between streams and applications.

Inter-agent communication facilitates inter-agent *cooperation* the goal of which is to provide the QoS levels requested by the application. An interesting new feature, compared to other QoS management schemes, which becomes possible due to this decentralized approach, is the possibility of communication between users resp. their local QoS agents, allowing for a cooperative selection of desired qualities. If users cooperate and decide to request a service in the same quality, less resources have to be reserved, which in turn leads to lower communication costs and higher resource availability for other applications. Details on our scheme can be found in (Fischer et al. 1997).

3 APPLICATION DESIGN

In (Hafid and von Bochmann 1997), we developed a framework for QoS management in the context of presentational multimedia applications. The multicast nature of many conversational applications, the participation of a huge number of users and the use of Cooperative QoS Management calls for a revision of this framework. We identified a number of major changes which

mainly affect two application components: a user-visible part, namely the QoS interface, and an invisible part, namely the QoS agent.

- QoS user interface

In our framework for presentational multimedia applications, users resp. QoS managers do not have the information on all existing multimedia document variants, and they access documents in individual sessions. The QoS user interface was then designed to give users the ability to specify the desired quality of service for each media type (audio, video, etc.) and thus to limit the number of document variants to be checked and offered by the QoS manager. As we already pointed out at the beginning, we believe that in multicast applications, only a limited number of qualities should be offered for a given media type. Using Coop. QoS Management, the information about all available media types and qualities is available from the network QoS agents. Thus, we do not ask the user to specify his requirements in advance, but offer him all available qualities for the session components. He then simply selects the ones desired.

- QoS agent

Our QoS manager in presentational multimedia applications goes through several consecutive operations to implement the negotiation process between a client and a server. This process involves different interactions between the QoS manager and the user on one hand, and the QoS manager and the remote service providers on the other. When the number of users becomes large, as in our teleteaching application, this process becomes too heavy for the whole system, since every single component would be in steady negotiation processes with new users. Due to the multicast nature of the application and the characteristics of Cooperative QoS Management, however, the design of the QoS manager can be simplified by far. If a certain stream is already supported in the area of the new user, then the QoS negotiation process need not be carried out; rather, the stream is simply multicast to the new user. Thus, an end system QoS agent has only to deal with QoS management on its own system. All issues concerning negotiations with, resource reservations at and adaptation of remote components are part of the work of the network QoS agents. End system QoS agents simply send messages to their neighboring agents, stating the types and qualities of documents/media they would like to receive.

The teleteaching application we use as an example supports the delivery of a lecture from a given site to students located in several remote locations. The delivery consists of video and audio from the lecturer. In addition, the lecturer may present multimedia documents stored locally or in some other locations. Students have the possibility to ask questions, but they first have to get permission to do so. In this prototype, we allow only one student to talk at a time. The lecturer is always allowed to talk. In addition, it is possible to record

Figure 1 Architecture of a receiver's machine

a whole session and store it as a multimedia document in the multimedia databases. Thus, lectures may be later reviewed by students when preparing for exams. We decided not to build a new application from scratch, but to use the existing MBone tools (Kumar 1996) and protocols *in combination with* our QoS user interfaces and protocols (Hafid and v. Bochmann 1996) (however modified as described above). The architecture of a receiver's end system can be found in Figure 1. In the current version of the application, a quality switch for a given medium is implemented by stopping the corresponding MBone tool and starting it again with a different multicast address. We plan to change the MBone tools' implementation to perform such a switch internally in order to provide a smoother quality switch.

REFERENCES

- Fischer, S., Hafid, A., v. Bochmann, G. and de Meer, H.: 1997, Cooperative Qos Management in Multimedia Applications, *4th IEEE Int. Conf. on Multimedia Computing and Systems (ICMCS'97)*, Ottawa, Canada. To appear.
- Hafid, A. and v. Bochmann, G.: 1996, Quality of Service Negotiation in News-on-Demand Systems: An Implementation, in A. Azcorra (ed.), *Proc. 3rd Int. Workshop on Protocols for Multimedia Systems, Madrid, Spain*, pp. 221–240.
- Hafid, A. and von Bochmann, G.: 1997, A General Framework for Quality of Service Management. Submitted to Computer Communications, Special Issue on Building Quality of Service into Distributed Systems.
- Kumar, V.: 1996, *MBone – Interactive Multimedia on the Internet*, New Riders Publishing, Indianapolis, Indiana.
- McCanne, S., Jacobson, V. and Vetterli, M.: 1996, Receiver-driven layered multicast, *ACM SIGCOMM'96*, Stanford, CA.