A new flexible and modular QoS mapping framework based on psychophysics

C. Gbaguidi, O. Verscheure, and J.-P. Hubaux
Telecommunication Services Group (TSG)/ TCOM Laboratory
Swiss Federal Institute of Technology (EPFL)
CH- 1015 Lausanne, Switzerland
e-mail: {gbaguidi, verscheure, hubaux}@tcom.epfl.ch
WWW: http://tcomwww.epfl.ch/~{gbaguidi,verscheu,hubaux}

Abstract

Accounting for the perceiving conditions that make up the delivery environment of an application helps improve the *efficiency* of QoS provisioning systems. It is useless to transmit information whose absence cannot be noticed by the end-user under the actual perceiving conditions. Until now, QoS architectures mostly focused on the transport system and did not integrate the studies achieved in the psychophysics area. In this paper, we propose a QoS framework that accounts for the perceiving conditions. Moreover, our framework is both *flexible* - i.e., customizable at will by the manager - and *modular*, with a clear and fine-grained layering. New mechanisms and their supporting characteristics, such as experimental curves, may be very easily introduced and managed in the proposed architecture. A case study is carried out, which shows the applicability of the framework for a video-on-demand provisioning system.

Keywords

Quality of Service (QoS) framework, perceptual QoS, psychophysics, QoS mapping.

1 INTRODUCTION

Mapping the Quality of Service (QoS) from the user's view down to the network infrastructure is one of the key issues to be addressed with the provision of multimedia services. This mapping is useful in many respects. *First*, it is helpful for optimizing resource allocation both in the end-systems and within the network, meaning a (near-)optimum estimation of the amount of resources to be dedicated to a given task. This is actually the major goal of QoS mapping. The estimated amount of resources is compared with the available resources which may be insufficient for the task to be performed; if this is the case, the mapping fails. Since the parameters to be considered in the mapping are numerous, many mapping rounds may take place before the optimum is found. *Second*, once the mapping has determined the part of the global QoS to be fulfilled by each resource

involved in the communication, QoS monitoring may be very easy and, finding out the QoS bottle-necks (resources that are not meeting the negotiated QoS) may be dramatically simplified.

An efficient and comprehensive QoS mapping framework must take into account the end-user's perception in order to optimize the amount of resources to be used. It is useless to transmit data that the end-user is unable to notice (*perceptual redundancy*). In the video coding area, high compression ratio may be reached by eliminating the perceptual redundancy as much as possible, in addition to reducing the spatial and temporal redundancies. For a given amount of bits to be allocated to a video sequence, the repartition of these bits among and within the pictures may be very different whether the user perception is considered or not. This knowledge can be advantageously used in the case of the transmission of video over packet networks which may induce packet loss. Although it is still very difficult to predict, the impact of data loss may be analyzed from the user perspective thanks to a perceptual video quality metric (i.e., a quality metric based on a complete spatio-temporal model of the human visual system (van den Branden Lambrecht, et al., 1996)). This introduction of the vision science knowledge into the video coding and transmission domain may lead to very interesting results (Verscheure, et al, 1996a-b),(Garcia, et al., 1996). The existing QoS mapping schemes, essentially those described in (Nahrstedt, et al., 1996) and (Coulson, et al., 1995), do not address this issue.

In this paper, we essentially intend to bridge the results from (Nahrstedt, et al., 1996) and (Coulson, et al., 1995) with studies carried out in both the vision science and video coding areas in order to build up an efficient, flexible and modular QoS mapping framework. *Efficiency* is improved by considering the user's perceptual features. By *flexibility* and *modularity*, we mean the ability of the framework to be easily customized by the application designer who is allowed to select the framework components (*modules*) that best match its intentions.

Our interests herein are *twofold*. *First*, we aim at describing the information needed for achieving the mentioned mapping. The resulting information model will exhibit the parameters which are relevant to the user perception of video. *Second*, we intend to provide a functional model which shows the actual negotiation of QoS among the entities involved in the communication session. This functional model will introduce some of our early experimental results in the investigation of the impact of vision science on video coding and transmission applications.

The aforementioned interests will be addressed as follows. In Sec. 2, the QoS layers across which the mapping takes place are described as part of a QoS architecture under development in our laboratory. The information model needed for describing the mapping of QoS parameters across the QoS layers is presented in Sec. 3. Section 4 describes some functional components needed within the QoS architecture. A complete case study related to the transmission of MPEG-2 streams over ATM networks is dealt with in Sec. 5. Finally, concluding remarks are given in Sec. 6.

2 A QoS FRAMEWORK

Figure 1 depicts the QoS architecture in support of all the work described in this paper. This architecture exhibits 8 layers and 4 planes. The layers describe the processing steps involved in communications between pairs of end-systems. The planes are projections of the QoS framework with respect to specific concerns. Hence, the information collecting and management plane represents the information needed to build up the framework, especially the estimation of parameters that influence QoS delivery. Planes and layers are described in turn in the following subsections (2.1 and 2.2). An overview of closely related works is given in subsection 2.3.
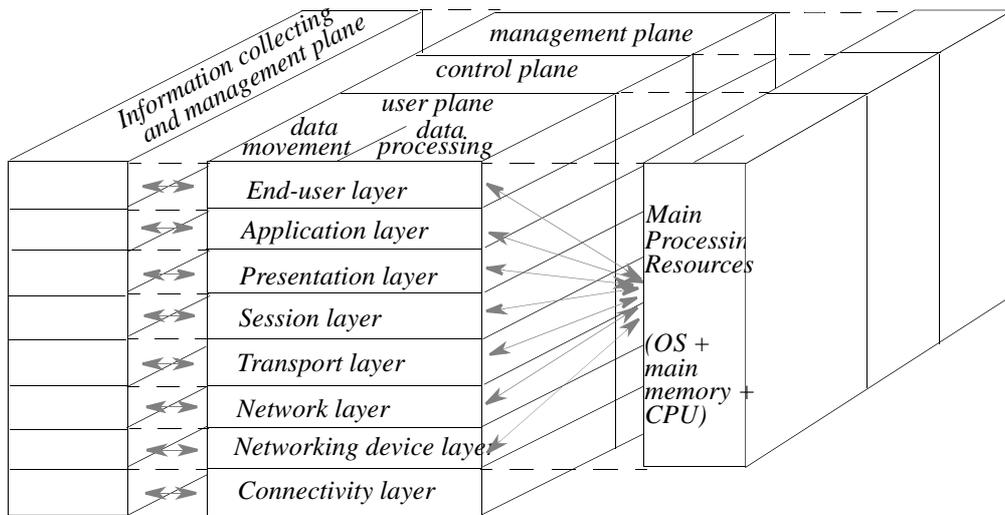
Figure 1 A QoS framework.

2.1 The QoS framework layers

The complexity of information processing between end-parties is broken down into 8 layers, namely the end-user, application, presentation, session, transport, network, networking device, connectivity. The main processing resources entity represents the resources provided by the host.

The *end-user layer* is concerned with the end-user's features and the facilities used for capturing the user's requests. Perceiving conditions are the most important features at this layer. They describe the actual user environment, e.g., in terms of the distance between the end-user and the screen, the surrounding luminosity, etc. The conditions related to video perception are defined in the ITU-R Rec. 500-3 (ITU-R 500-3, 1986).

The *application layer* deals with a first refinement of the information carried in from the end-user layer. This refinement process essentially confronts the user's QoS demands with its perceptual features and the selected type of application (videoconferencing, teleteaching, video-on-demand, etc.). The QoS parameters to be requested from the lower layers are then derived by the refinement process.

The *presentation layer* contains the devices needed for capturing and rendering information in a meaningful shape for the end-user. The presentation concerns include information capturing/displaying, inter-media synchronization, and compression/decompression. Therefore, the presentation layer is the last in the order of our description which is directly influenced by the end-user's perceptual features.

The *session layer* serves as a coordination layer between the presentation and application layers, and the transport system. This coordination layer is usually implemented by the so-called sockets.

The *transport layer* controls the transfer of information on an end-to-end basis.

The *network layer* implements functions that shield the upper layers from the discrepancies among network technologies (ATM, Ethernet, FDDI, etc.).

The *networking device layer* may relate both to the Network Interface Card (NIC) and any other similar device inside the network. The activities achieved by this layer are tightly dependent on the underlying network technology.

The *connectivity layer* abstracts over the connectivity between networking devices. This connectivity may be conferred different degrees of abstraction as illustrated in Figure 2. In case the network is considered as a "black box" (*implicit connectivity*), the connectivity layer represents the

end-to-end connectivity service offered by the network infrastructure. The layer in question may also abstract over a physical link segment between two adjacent nodes (*explicit connectivity*). This detailed view of the connectivity may be used by the network control system to distribute the QoS required from the network over the switches or routers or other elements along the path between the end-systems involved in the communication.
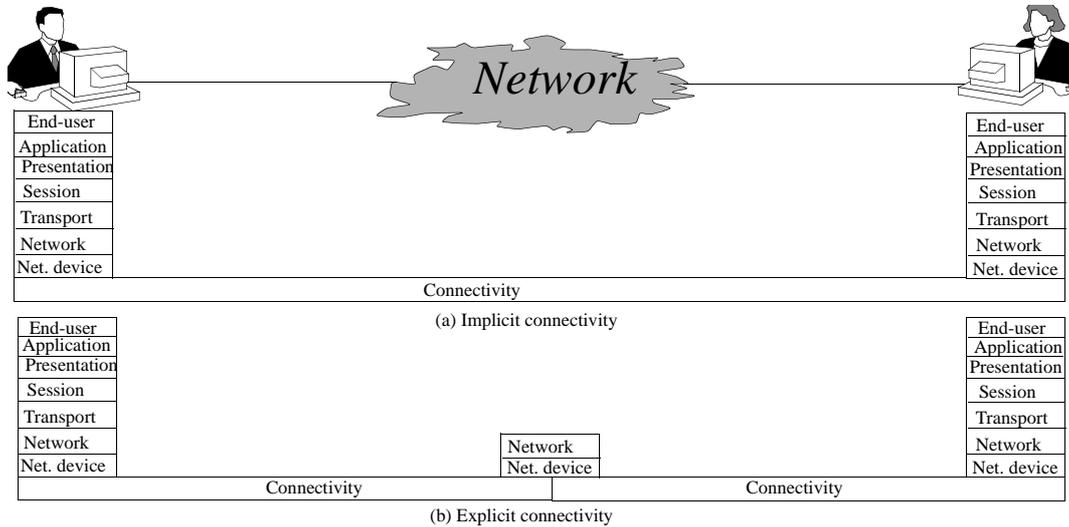
Figure 2 Example of abstraction degrees encompassed by the connectivity concept.

The *main processing resources* entity provides resources for the execution of the functions incorporated in layers that lack own processing capabilities. This is generally the case of layers implemented in software such as the session, transport and network layers. The main processing resources entity considers the actual implementation of the communication architecture in the end-systems. More importantly, it shows that two layers may not be using the main processing resources at the same time. This situation must be accounted for when mapping the delay introduced by the end-system in an end-to-end communication.
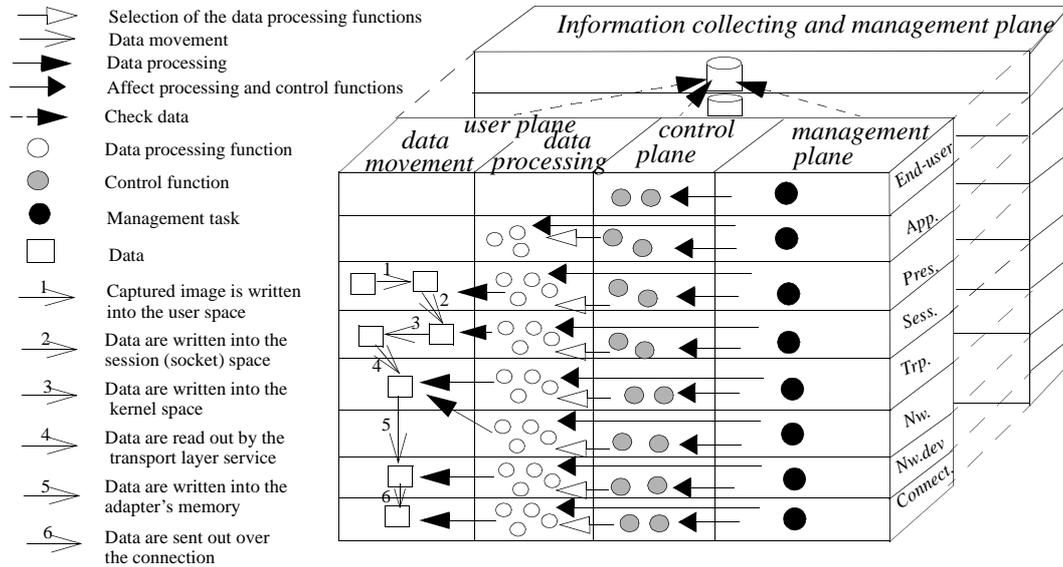
2.2 The QoS framework planes

Each plane abstracts over the global framework with respect to specific purposes. Four planes which cover different perspectives are considered: the user plane, the control plane, the management plane, and the information collecting and management plane.

As depicted in Figure 3, the **user plane** is concerned with media transfer and it serves *two* purposes. *First*, it provides the data path through the end-system (data movement sub-plane). In Figure 3, an example is given, where an image is first captured by a camera, and then copied into the user space. The session layer copies the image file into its memory space and waits until the transport layer is ready to process new data. Then, the session layer writes the data into the transport layer memory space. After all the processing related to the transport and network layers is performed, the data is written into the network adapter's memory from where it will be transmitted over the connection set up for the purposes. *Second*, the user plane controls the media transfer across the layer by performing so-called protocol functions (data processing sub-plane). Generic functions are: sequence control, flow control, congestion control, replicate control, data corruption control, acknowledgment, retransmission, error detection, error correction, segmentation, reassembly, stream synchronization, and encoding/decoding as outlined in (Zitterbart, et al., 1993).

The choice of the functions to be performed for a given connection is made by the **control plane**. In other words, the control plane configure the user plane to suit the needs of the connection.

The control plane is used for negotiating or re-negotiating QoS, and for performing other complex tasks with the support of the user plane, especially the data processing sub-plane. The control plane performs *two kinds of mapping*. The *first* mapping consists in selecting the data processing functions that best suit the needs of the service, and, *second*, a mapping takes place for QoS parameters. A way these mappings may be performed is described in Sec. 4.

Figure 3 Interactions among the framework planes.

The ***management plane,*** as illustrated in Figure 3, is used for affecting the parameters associated with the control and data processing tasks. For instance, the parameters of the layer admission control algorithm may be changed by the layer management entity. These parameters may be the amount of resources to be used, the number of simultaneous connections, the traffic classes to be considered, etc.

The ***information collecting and management plane*** monitors the factors that influence QoS and provides mechanisms for estimating these factors. The estimation results are used by the user, control and management planes. The information collecting and management plane may also be used for storing management information which may be accessed by means of the conventional management protocols, specifically SNMP (Simple Network Management Protocol) (Case, et al., 1996) and CMIP (Common Management Information Protocol) (ITU-T X.711, 1991). The introduction of the information collecting and management plane has been motivated by the heterogeneity among end-systems. The way factors that influence QoS are estimated may be different from one system to another, depending on the system's devices. Therefore, separating this estimation from the other planes improves the flexibility of the framework as to cope with differing environments. Moreover, the plane of concern may be used for influencing the QoS mapping process as will be explained in subsection 3.3 and more concretely settled in the case study in Sec. 5. The information collecting and management plane plays a similar role as the D-plane of the binding architecture described in (Lazar, et al., 1996).

2.3 Related work

A relevant review of QoS architectures has been achieved in (Campbell, et al., 1996). For the sake of conciseness, all these architectures are not mentioned here. However, it is worth noticing that *two*

of them - specifically the OSI reference model (ITU-T X.200, 1994) and QoS-A from Lancaster University, UK, (Campbell, et al., 1994) - show many architectural similarities with the framework proposed in this paper.

Our framework may relate, at first sight, to the OSI model (ITU-T X.200, 1994). However, the two architectures do differ in their underlying philosophies and layerings. The OSI philosophy consists in describing the interfaces between the layers exhibited by the architecture, some of which are termed the same in our architecture. On the other hand, our framework depicts a QoS mapping scheme which quantifies the layers in terms of their induced delay, probability of loss, jitter, etc. Our philosophy is rather motivated by the provision of a simple way to estimate the QoS to be requested from the layers. For QoS mapping purposes, the important is the QoS to be requested from both the end-systems and the network. Splitting the end-systems activities into layers as we did just helps cope easily with different end-system capabilities.

With regard to QoS-A, its layering is very similar to the OSI reference model (ITU-T X.200, 1994), except that the 3 upmost layers in the latter model have been replaced by two layers called orchestration and distributed systems platform layer. The orchestration layer essentially deals with synchronization among several (media) connections belonging to the same communication session. In other words, the orchestration layer is concerned with inter-media synchronization. The distributed systems platform layer implements the functionalities needed for coordinating interactions among a set of end-systems. QoS-A addresses most of the mechanisms needed within a QoS framework.

The major difference between QoS-A and our framework lies in *three facts*. *First*, we consider the main processing resources as a whole entity. Therefore, the mechanisms in support of real-time requirements easily fit into our framework and are unambiguously stressed. *Second*, we feel that inter-media synchronization should be dealt with at the very last processing steps. Therefore, inter-media synchronization should be comprised by the presentation layer (in our framework) which is in charge of decompressing and displaying the information. *Third*, and more importantly, the end-user's perceptual features are taken into account in our framework, which have not been addressed in QoS-A yet.

In this Section, we have described the framework in which all of the following results will fit. At this point, any factual evidence of the use of results from the psychophysics domain has not been clearly given yet. A first step towards this evidence is achieved in the ensuing Section through the modeling of QoS related information at the layers to which the end-users perceptual features are directly influencing.

3 AN INFORMATION MODEL FOR THE QoS FRAMEWORK

In this section, we focus on the information necessary for taking into account the user's perception of the delivered QoS. Obviously, the influence of the user's perceptual features will "propagate" from the end-user to the presentation layer. The session layer as well as the layers below are not directly affected by those features. Therefore, we devote this section to organizing the information that influence the mapping of the QoS from the end-user view down to the presentation layer.

3.1 Modeling the end-user layer

As depicted in Figure 4, the end-user layer aggregates the end-user and the layer controller (*EUL_Controller*). The end-user's perception is influenced by the perceiving conditions that describe the actual conditions in which the user wants the application to be provided. Perceiving conditions are essentially composed of viewing conditions - which influence the perception of video information - and hearing conditions which affect the perception of audio information.

The end-user also may select the media that he or she is willing to be provided with. These

media may need to be synchronized with one another. Moreover, each medium is constrained by QoS requirements expressed by the end-user in terms of a QoS level, generally in range 1 to 5 as recommended in (ITU-R 500-3, 1986) for video services. A *Stats* object collects statistics about the delivered QoS. The overall cost that the user is willing to incur for the application is imposed thanks to the *Cost* object.

The controller is in charge of supervising the activities at the end-user layer. It is supported by some functions such as *UserInterface*, which captures the user's desiderata, and *ParametersConsistency*, which checks the consistency of the parameters supplied at the layer before any further request is issued to the layer below, for instance. The layer functions may call one another and ask for assistance from the application layer.

Note that we represent functions here as objects, which may shock purists. We feel that functions can be updated, changed, triggered, deactivated at will by the user or the manager. Therefore, representing functions as objects makes sense.
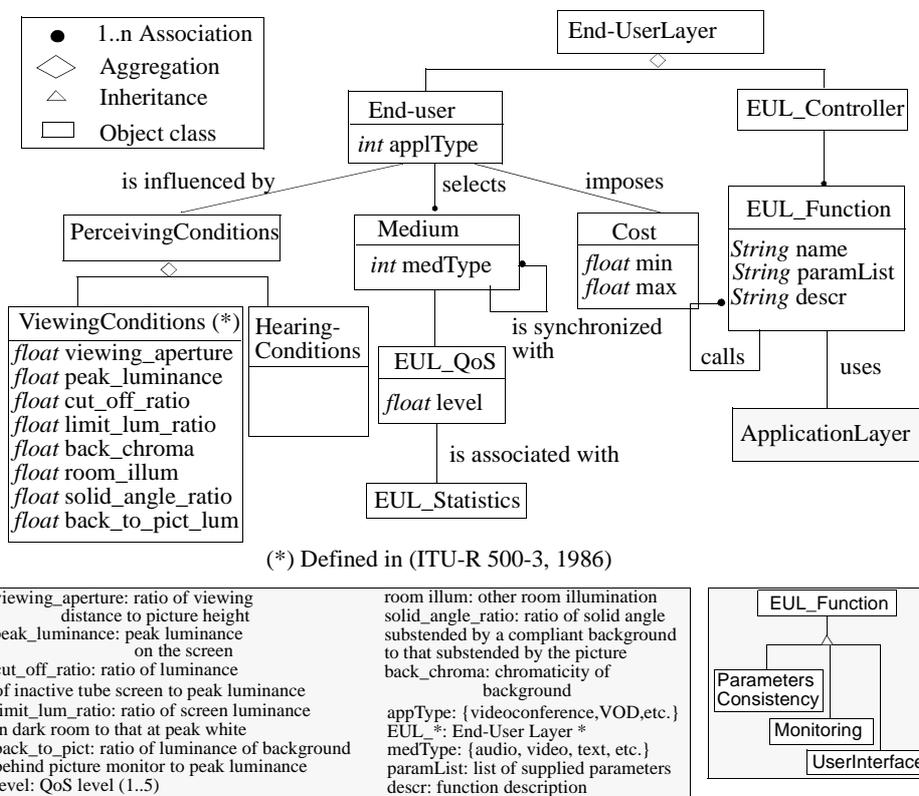
Figure 4 A modeling of QoS-related information at the end-user layer.

3.2 Modeling the application layer

The application layer performs a first refinement of the information supplied by the end-user layer. Each layer may incorporate many application objects which relate to the end-user's purposes about the service to be provided. Examples of application are videoconferencing, teleteaching, conversation, cooperative work, etc. The provision of these applications is possible thanks to underlying services which are application-independent. Hence, the telephony service may be used for both conversation between end-users, and data transfer between two computers. In other words, the same service may serve several purposes known as applications.

Each application involves a number of media which are coerced by QoS constraints. In the case of video applications, these constraints apply to regions of interest determined either by the end-user or automatically by a dedicated function in the system. For instance, the region of interest during a videoconference may be the faces of the conferees. The QoS requested for this region should be more stringent than the one for the other regions. Automatic detection of regions of interest is intended for MPEG-4 codecs (ISO-IEC/JTC/SC29/WG11 N1022, 1995). Regions of interest are characterized by their shape (rectangle, circle, etc.), and their associated QoS in terms of the minimum level (in range 1 to 5), the average level over the time of the communication, and the degradation burst size. The latter metric denotes the maximum period of time over which maximum QoS degradation (minimum QoS level) can be tolerated. We have introduced this metric to alleviate drawbacks of using statistics, which only describe a relatively long-term vision of situations. Particularly as to QoS concerns, this kind of vision is too lacking, since the end-user is very sensitive to burst of degradation and it will remain unsatisfied even if the average QoS negotiated has been delivered.

As in the case of the end-user layer, the application layer contains a controller which is responsible for deploying the appropriate functions in response to requests from the adjacent layers.
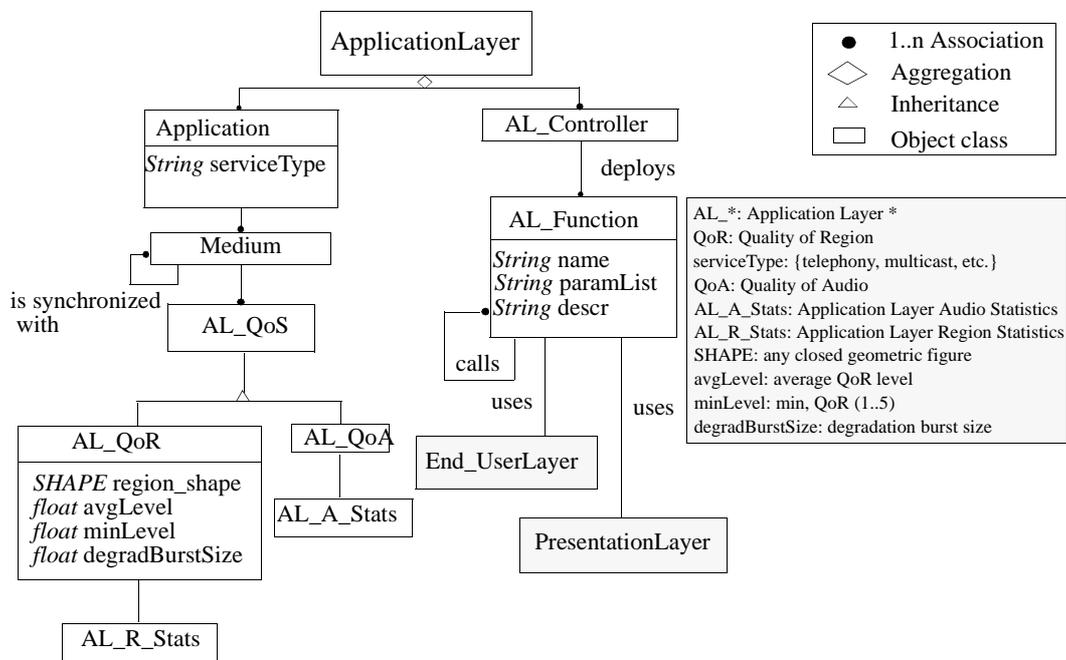

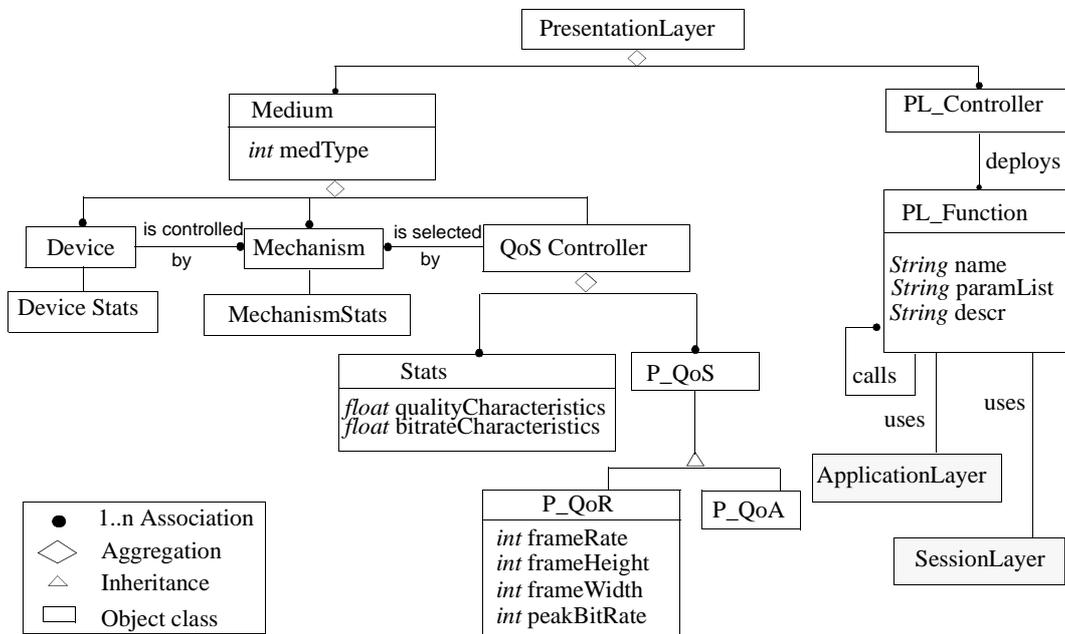**Figure 5** A modeling of QoS-related information at the application layer.

3.3 Modeling the presentation layer

The presentation layer is concerned with accommodating the information to the end-user's perception. It aggregates the media required by the application layer and a controller which plays the same role as in the layers described above. Each medium aggregates the adequate processing devices, a medium QoS controller, and some mechanisms. We believe that *four main groups of mechanisms* may be designed in order to control the *encoding process of a video source*: the rate control, the quality control, the robustness and the tools based on psychophysics groups (see (Verscheure, et al, 1996b) for examples). The *rate control* mechanism group aims at regulating the encoded video stream in order to produce either a constant or a controlled-variable bit rate stream. This group of

mechanisms optimizes the user requirements in terms of the traffic characteristics. It also helps the video source comply with the traffic characteristics negotiated with the network, thus avoiding traffic violation by the source.

The video quality is controlled by the *quality control* group of mechanisms, which may, for instance, be requested to produce a constant video quality over time. In the case of an error-prone environment, one may wish to increase the robustness of the encoded video stream. The *robustness* group of mechanisms aims at reducing the impact of an error on the video quality of the reconstructed video sequence by adding syntactic redundancies (e.g., by reducing the slice size in the case of MPEG-2 encoding (ISO-13818-2, 1994)). Each of these groups may or may not use the *tools* based on psychophysics described in (Verscheure, et al, 1996b), specifically video quality and activity metrics based on a complete model of the human visual system.

The mechanisms are selected by the QoS controller and they affect the devices activities. The QoS controller object aggregates many statistics objects and many presentation layer QoS objects, which may represent video QoS or audio QoS. As mentioned above, the video QoS object is an aggregation of Quality of Region (QoR) objects. QoR is characterized here by the frame rate, frame size, color depth, etc.

Figure 6 A modeling of QoS-related information at the presentation layer.
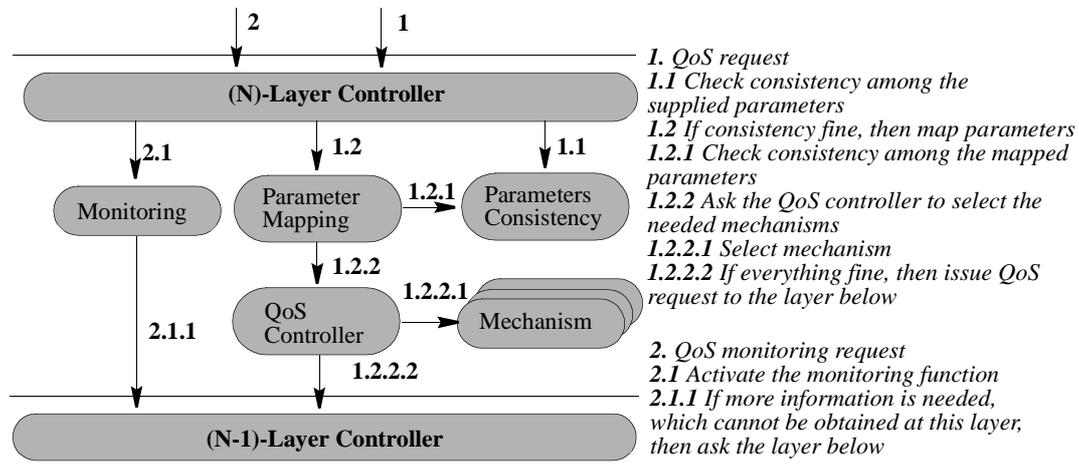
In this Section, we have described the information needed for computing and satisfying QoS at the three layers directly affected by the end-users perceptual features. In the next Section, we are going to describe how QoS satisfaction may be functionally achieved within our framework.

4. A FUNCTIONAL MODEL FOR THE QoS FRAMEWORK

The goal of this Section is to show how the framework processes QoS demands. We will particularly explore simple ways to perform control and management operations within our architecture.

Figure 7 depicts the processing of a control request issued by layer (N+1) to layer (N). With respect to the framework in Figure 1, Figure 7 maps to the processing carried out within the control plane. The layer controller, which receives the request, first checks the parameters passed in by

layer (N+1) for correctness and consistency. In case this checking succeeds, a parameter mapping is performed to determine the correspondence between the parameters supplied by layer (N+1) and those relevant to layer (N). After the mapping is achieved, another consistency check is performed for the same reasons as aforementioned. Then, the QoS controller block is called to identify the mechanisms corresponding to the parameters that came out of the mapping process.

1. *QoS request*
1.1 *Check consistency among the supplied parameters*
1.2 *If consistency fine, then map parameters*
1.2.1 *Check consistency among the mapped parameters*
1.2.2 *Ask the QoS controller to select the needed mechanisms*
1.2.2.1 *Select mechanism*
1.2.2.2 *If everything fine, then issue QoS request to the layer below*

2. *QoS monitoring request*
2.1 *Activate the monitoring function*
2.1.1 *If more information is needed, which cannot be obtained at this layer, then ask the layer below*

Figure 7 A processing model for control requests issued to layer (N).

As will be explained in Sec. 5, the *ParameterMapping*, *Mechanism*, and *QoS controller* blocks are likely to use experimental curves stored in the information collecting and management plane. These curves as well as the functions and mechanisms in each layer may be changed, updated, and introduced. These operations define management facilities that should be offered by the QoS framework. Figure 8 depicts the sequence of actions that implement these facilities. With respect to the framework in Figure 1, Figure 8 maps to the processing carried out within the management plane. Management requests are issued to the layer manager which triggers the checking of the consistency of the information supplied through the request. Since there may be some relationships among the curves, mechanisms and functions supplied, a *Curve-Mechanism-Function* association block is called to bind the elements of concern to each other. For instance, when a new curve is introduced, one has to define the mechanism or function with which it should be associated. The layer binds the curve to the mechanism or function of interest and notifies the layer controller and the QoS controller of the change. The layer controller needs to be aware of the functions available in the layer, and the QoS controller needs to know about the mechanisms to be triggered and their characteristics as well.

In this Section, we have essentially described one way the QoS control and management requests can be processed within a layer. This functional description is used in the next Section with a concrete example, specifically a video-on-demand application. The steps that lead to the choice of the mechanisms to be performed are outlined.
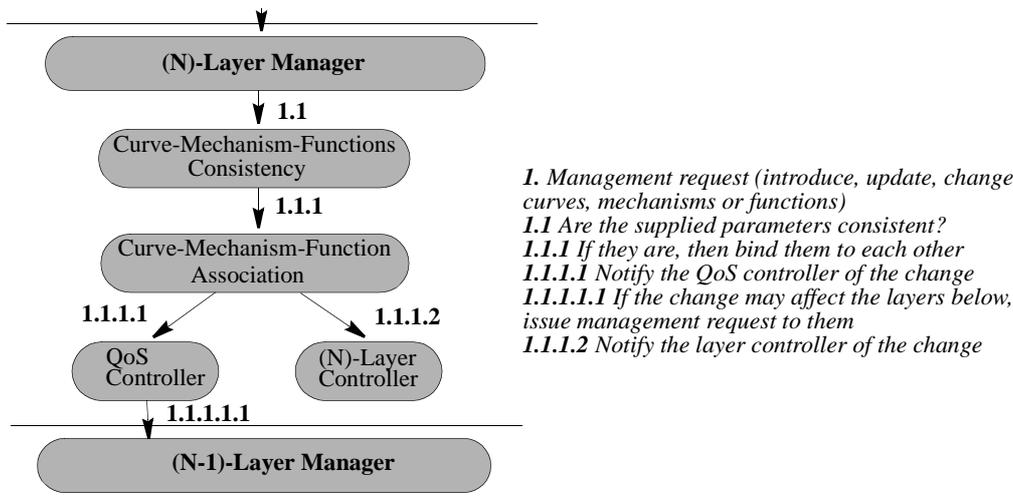
Figure 8 A processing model for management requests issued to layer (N).

5 CASE STUDY: A VIDEO ON DEMAND APPLICATION

This section deals with the description of a concrete example related to the transmission of MPEG-2 video streams over terrestrial ATM networks. We actually aim at finding the best trade-off between the perceptual video quality and the service cost for a video on demand (VoD) application. In other words, we consider that the user imposes a quality level but does not specify the cell loss ratio nor the encoding bit rate. The stream supplied by the video server is assumed to be scalable, so that different levels of quality may be available. We show now the decisions the system has to make in order to map this user requirement onto inputs to be supplied to the control mechanisms (Figure 9). Also, the system should be able to send back to the end-user application the bandwidth and the cell loss ratio to be negotiated with the network, as well as the price of the proposed session. Moreover, since the user does not supply any information about the perceiving conditions, these are set to their default values as recommended in (ITU-R 500-3, 1986).
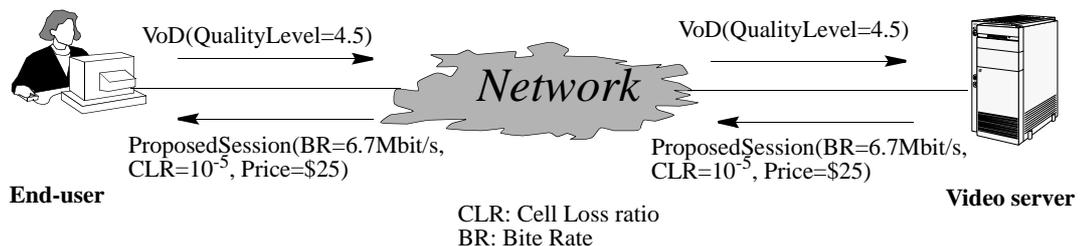

Figure 9 A video on demand provisioning system.

We assume that the video on demand system conforms to the specification proposed by the ATM Forum in (ATM Forum, 1996). This specification recommends the transmission of MPEG-2 Single Program Transport Streams (SPTS) at a constant packet rate (CPR) over the ATM Adaptation Layer 5 (AAL5).

The implications of delivering CPR are *twofold*. *First*, the QoS controller has to select a rate control mechanism which regulates the server's output at a constant bit rate. Therefore, the only traffic descriptor that has to be determined is the peak bit rate (i.e., the encoding bit rate). However, this constant bit rate encoding does not allow for the selection of the quality control mechanism which maintains the quality over time at a desired level. *Second*, delivering CPR implies that a very small cell loss ratio is expected from the terrestrial ATM network. Moreover, it has been shown in (Garcia, et al., 1996) that a CLR up to 10^{-5} did not introduce any perceptual degradation in the reconstructed video sequence, provided some basic error concealment techniques are used. This means that no robustness mechanisms need to be selected by the QoS controller, according to the functional model in Sec. 4.

In a word, only *one mechanism* is needed for providing the VoD application considered in this case study. This mechanism is the *rate control* which is requested to produce a constant bit rate at the video server's output. In the following paragraphs, we describe the way the selected mechanism may be deployed, and we identify the parameters to be supplied to it.

The QoS mapping function has to find the best trade-off between the perceptual video quality and the service cost using information - specifically empirical curves - stored in the information collecting and management plane (IC&M) (see subsection 2.2). Figure 10 illustrates an empirical curve showing the quality rating versus both the constant encoding bit rate and the cell loss ratio in the case of a VoD application over an ATM network using AAL5 (see (Verscheure, et al., 1996b) and (Garcia, et al., 1996) for more details).
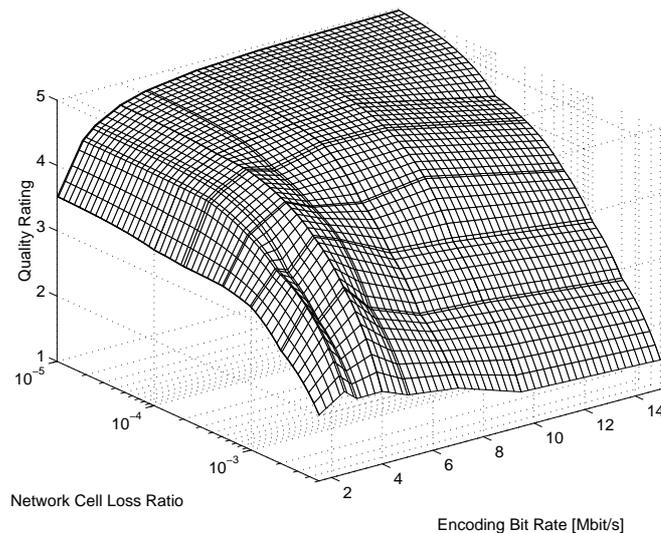

Figure 10 Empirical curve showing the video quality vs. the bit rate and and the cell loss ratio.

The CLR having been set to 10^{-5}, the operating point is determined in the derived two-dimensionnal graph represented in Figure 11 and stored in the IC&M plane as: $q = 4.9(1 - e^{-a(r + b)})$ where q denotes the quality level, r the encoding bit rate, and a and b two parameters which, in our case, evaluate to 0.45 and 0.2, respectively. Given that curve, we can easily find the operating point.

Let us define the operating point as the encoding bit rate above which the difference in quality is less than or equal to 0.1 for an increase of the bit rate of 1 Mbit/s. According to that definition, the optimal encoding bit rate is given by: $-\left(\frac{1}{a}\ln\left(\frac{1}{49a}\right) + b\right)$, which in our case yields 6.7 Mbit/s. The corresponding price is assumed to be \$25. Hence, the session proposed to the user is characterized

by the bit rate (6.7 Mbit/s), the cell loss ratio (10^{-5}), the price ($25), and the quality level (4.5). Upon receipt of this session descriptor, the user makes the decision whether he or she wishes to see the movie or not. Another alternative to avoid many price negotiation rounds would be for the user to specify the price he or she is willing to incur for the provision of the service. On this basis, the server would find the optimum operating point for the session. This supposes a pricing scheme which we have not studied yet; this is the reason why that alternative has not been considered.
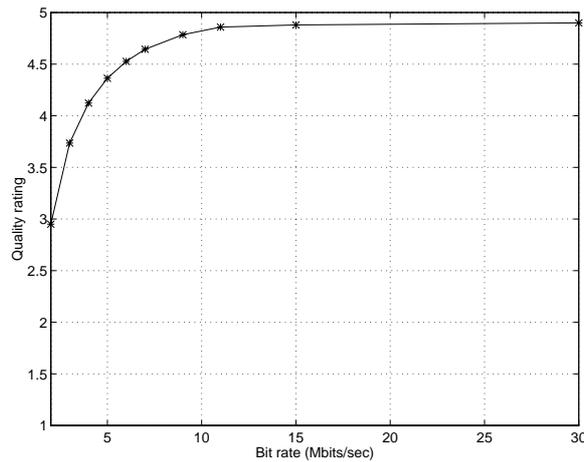
Figure 11 Focusing Figure 10 for CLR = 10^{-5}.

6 CONCLUSION

In this paper, we have investigated the introduction of results obtained in the psychophysics area into the QoS delivery system. We have proposed a QoS framework which abstracts over this system, yet with a tight connection with the actual processing steps undergone by the information conveyed between two end-users. The framework is composed of 8 layers - the end-user, application, presentation, session, transport, network, networking device, connectivity layer - and 4 planes - the user, control, management, and information collecting and management plane. The perceiving conditions which affect the QoS perceived by the end-user are taken into account within the end-user layer. The influence of those conditions will propagate directly down to the application and presentation layers. The other layers are not directly affected by the perceiving conditions. Therefore, we have focused on describing the information to be controlled and managed at the end-user, application and presentation layers. We have also proposed a functional model which depicts the processing of control and management requests issued to a generic layer (N). A scenario of integrated use of the information and functional models has been studied in the case of a video-on-demand application.

As future issue, we are investigating the mapping of the QoS parameters across the layers of the framework. Moreover, the implementation of the entire set of mechanisms to be selected at the presentation layer in order to cope with the perceiving conditions is on the way. These mechanisms are essentially rate control, robustness, and quality control.

7 ACKNOWLEDGMENTS

REFERENCES

The ATM Forum. (January 1996) Video on Demand Specification 1.0.

van den Branden Lambrecht, C., and Verscheure, O. (February 1996) Perceptual Quality Measure using a Spatio-Temporal Model of the Human Visual System. *IS&T/SPIE*, San Jose, USA.

Campbell, A.T., Coulson, G., and Hutchison, D. (April 1994) A Quality of Service Architecture. *ACM SIGCOMM Computer Communication Review.*

Campbell, A., Aurrecoechea, C., and Hauw, L. (March 1996) A Review of QoS Architectures. *4th International IFIP Workshop on Quality of Service,* Paris, pp.173-195.

Coulson, G., et al. (1995) Design of a QoS Controlled ATM Based Communications System in Chorus. *IEEE Journal of Selected Areas in Communications (JSAC),* Special Issue on ATM LANs Implementation and Experiences with an Emerging Technology, 1995.

Garcia Adanez, X., Verscheure, O., and Hubaux, J.-P. (Nov. 1996) New Network and ATM Adaptation Layers for Real-Time Multimedia Applications: A Performance Study Based on Psychophysics. *3rd COST 237 International Workshop on Multimedia Telecommunications and Applications*, Barcelona, Spain.

Case, J., et al. (January 1996) Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2). IETF RFC 1905, Draft Standard.

ISO-13818-2 (1994) Information Technology- Generic Coding of Moving and Associated Audio Information- Part2: Video. Draft International Standard ISO/IEC/ DIS 13818-2.

ISO-IEC/JTC/SC29/WG11 N1022. (July 1995) Requirements for the MPEG-4 Syntactic Description Language (Draft Revision 2). Tokyo, Japan

ITU-R Rec. 500-3 (1986) Method for the Subjective Assessment of the Quality of Television Pictures.

ITU-T Rec. X.200. (July 1994) Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model. Also known as ISO/IEC 7498-1.

ITU-T X.711 (1991) Common Management Information Protocol Specification for CCITT Applications.

Lazar, A. A., Lim, K.-S, and Marconcini, F. (Sept. 1996) Realizing a Foundation for Programmability of ATM Networks with the Binding Architecture. *IEEE Journal on Selected Areas in Communications,* vol. 14, no. 7, pp1214-1227.

Nahrstedt, K., and Smith, J. M. (Sept. 1996) Design, Implementation, and Experiences of the OMEGA End-Point Architecture. *IEEE Journal on Selected Areas in Communications,* vol. 14, no. 7, pp1263-1279.

Verscheure, O., Basso, A., El-Maliki, M., and Hubaux, J.-P. (Sept. 1996-a) Perceptual Bit Allocation for MPEG-2 CBR Video Coding. *International Conference on Image Processing (ICIP)*, Lausanne, Switzerland.

Verscheure, O., and Hubaux, J.-P. (Nov. 1996-b) Perceptual video quality and activity metrics: Optimizations of video services based on MPEG-2 encoding. *3rd COST 237 International Workshop on Multimedia Telecommunications and Applications*, Barcelona, Spain.

Zitterbart, M., Stiller, B., and Tantawy, A. (May 1993) A Model for Flexible High-Performance Communication Subsystems. *IEEE Journal of Selected Areas in Communications (JSAC),* Vol. 11, no. 4, pp507-518.