Some Principles for Quality of Service Management

Gregor v. Bochmann and Abdelhakim Hafid

Université de Montréal
Dept. IRO, CP 6128, Succursale centre ville, Montréal, H3C 3J7, Canada
E-mail: {hafid, bochmann}@iro.umontreal.ca

Abstract: Distributed multimedia applications require a variety of levels of quality of service (QoS) from communication networks and end-systems which realize the multimedia interface with the human users or provide remotely accessed multimedia information. The management of these services has to take into account the available resources and the user's wishes concerning the desired quality and costs. Based on our experience with the construction of a News-on-Demand prototype, we present in this paper a few simple principles applying to QoS management for distributed multimedia applications. We discuss in particular ways in which an application can adapt to reduced network performance related to throughput, loss, delay or jitter, and we consider the situation where the configuration of the distributed application can be selected dynamically and be revised during the running of the application. We also comment on the handling of QoS in a layered system architecture, the use of performance models for QoS management.

Keywords: Quality of service, management, adaptation, negotiation, news-on-demand

1. Introduction

The notion of quality of service (QoS) has been used with various meanings. We consider in this paper the context of distributed multimedia (MM) applications, for which a possible definition of QoS is as follows: "Quality of service represents the set of those quantitative and qualitative characteristics of a distributed multimedia system necessary to achieve the required functionality of an application" [Vog 95]. It includes performance-oriented attributes, such as transmission delay and bit-rate, format-related attributes, such as video resolution, frame rate, storage format and compression scheme, synchronization aspects, such as skew between audio and video sequences, cost issues, such as connection, data transmission and copyright charges, and user-oriented attributes, such as subjective image and sound quality. Some overview of QoS issues in distributed multimedia systems was given in [Vog 95].

Issues of distributed systems management are presently of much interest because the systems to be operated become more and more complex and their management become a task which is difficult to perform. Much work has been done in the context of standardized frameworks, including models for objects to be managed and protocols to exchange information about managed objects within distributed systems. QoS management can be seen as a special aspect of distributed systems management. This area of management is concerned with finding appropriate QoS characteristics for the different system components in a distributed multimedia application and reserving the corresponding resources in such a way as to achieve the required functionality of the given application and to optimize the overall system performance.

There has been much work in recent years in the field of QoS management for communication networks. This work concentrated on resource allocation in communication networks in order to assure specific QoS characteristics for requested new connections. Much of this work is related to ATM networks which are expected to provide specific QoS guarantees if requested by the user. There have also been several proposals for including resource reservation schemes within the Internet in order to allow for some form of performance guarantees.

Another important area of research has been related to resource allocation within real-time operating systems in order to be able to guarantee specific performance guarantees for the real-time processing of multimedia data. An important application has been the development of video servers that are capable of delivering video streams in real-time to a large number of users which are normally connected through some communication network.

In this paper, we take a more global perspective. Resource allocation within the network, and within a computer operating system are considered two specific subproblems within the global system architecture involving in general many networks, computers, applications and users. The network and the continuous media file server, for instance, are only a few of the system components that together support the distributed multimedia applications of interest. In this global perspective, there are two other areas which are of particular interest, namely (1) the interface by which QoS characteristics can be negotiated with the user, and (2) the issues which are related to the adaptation of the application to the QoS characteristics available in the global system. Such adaptation may involve various forms of trade-offs, including alternative configurations and options, which may depend on the nature of the application at hand.

It is clear that the problem of resource allocation for QoS management, in general, is a very complex problem, for which no optimal and efficient solution exists. In this paper, we describe a few simple principles and concepts which seem to be useful for describing most of the issues that occur in QoS management within a global context. However, we do not cover the issues related to resource allocation in networks or computer operating systems. We hope that this presentation will be useful for high-lighting the key issues in QoS management and identifying the approaches and techniques that may be useful for solving some of the outstanding problems.

The paper is organized as follows. We begin by presenting an example application, namely News-on-Demand, which involves the access of users to remote multimedia databases. Section 3 discusses QoS in communication systems; more specifically it defines the notion of streams and their QoS-related characteristics. Section 4 presents some possibilities for adaptation of multimedia applications to reduced QoS availability. Issues related to the negotiation with the users are discussed in Section 5, and Section 6 outlines a global QoS negotiation framework which includes the consideration of alternative global configurations. A prototype implementation of the News-on-Demand application, introduced in Section 2, is discussed in Section 7, and Section 8 concludes the paper.

2. An example system: News-on-Demand

Multimedia information systems integrate diverse media such as text, video and images to enable a range of multimedia applications including information retrieval. The News-on-Demand application allows the user to browse through a remote database of news articles and retrieve selected articles for display on the workstation. A news article may consist of a single monomedia component, such as a text, an image, a video clip, or a sound track, or may be composed of a combination of such monomedia objects. The presentation of the continuous media, namely video and voice, is particularly sensitive to the available QoS in the network as well as in the server and the user workstation.

News-on-Demand has been selected as the target application within our collaborative research project "Broadband Services" funded by the Canadian Institute for Telecommunications Research (CITR). This target application embodies many features that are generally applicable to many multimedia presentational applications, such as digital libraries or computer-assisted training. In our current prototype, the user may choose a document in the database for presentation, and select the desired QoS including such parameters as video and audio quality, size of display, and cost. A graphical interface is available for this purpose which includes the possibility of viewing an example of specific quality features. The transmission of the continuous media components of the document, e.g. video and audio, proceeds in real-time over ATM or a local network during the presentation of the document. The system allows for several versions of a given media component, possibly with different QoS parameters and accessible from different servers over different networks. The QoS negotiation and adaptation features allow for the selection of the best configuration for a given user request and for automatic adaptation in case of changes of the QoS system parameters, such as in case of network or server congestion.

Within the high-level architectural design of the prototype [Boc 96], we have identified the following system components:

- The database manager allows to store and retrieve MM documents. It provides an interface which allows the QoS manager to get meta-information, such as the different variants and their locations and temporal relationships between related media, on a given MM document.
- The profile manager is in charge of managing the user's QoS preferences, that is, helping the user while setting and modifying his/her requirements through user profiles. The profile manager is also in charge of providing services to the QoS manager to find the profile selected by users.
- The network monitor allows to monitor the quality actually provided by the network for a given connection. When the effective presentation quality does not conform anymore to the target values selected during QoS negotiation, the network monitor sends a violation notification to the QoS manager.
- The QoS Manager performs the QoS negotiation and adaptation by interacting with various system components. That is, before running the negotiation protocol, the QoS manager gets (1) meta-information related to the document to be played from the database manager, (2) the agreed user profile from the profile manager, and (3) statistical information on the system (network) load from the network monitor. The QoS manager executes the adaptation protocol again during the presentation of the document when a violation notification is received.

As an example, let us consider a scenario involving QoS degradation. We assume that a presentation session is in progress and that the network becomes congested thus leading to lower QoS parameters for the transmission service. Let us assume also that the QoS manager has initiated network monitoring for the connection with threshold values selected to assure the worst acceptable QoS accepted by the user profile. If the measured value of a QoS parameter goes below the threshold, the monitor will notify the QoS manager. The QoS manager has essentially four choices in such a situation: (a) to continue the ongoing session with the present QoS parameters, (b) idem, but also to notify the application of the reduced QoS, (c) to automatically select another configuration, and (d) compute all possible configurations in order to let the application decide.

In the case of automatic reconfiguration (choice (c)), the QoS manager stops the presentation of the document after having determined the current position of the document. It then determines the best alternate configuration, activates this configuration and restarts the presentation from the position parameter determined earlier.

The subsequent sections of this paper discuss different aspects of QoS negotiation and use the above application as an example.

3. The quality of communication services

It is common sense to describe a complex system in terms of its components which in turn may be composed of more simple subsystems. Using an object-oriented specification framework, such as ODP [Ray 93], a subsystem may be considered an object characterized by attributes and operations. An operation may be called by another object, and the function performed during such a call can be characterized by a procedure which provides the desired results at the end of its execution. This is quite different from the nature of audio and video communication where the essence of the communication function is performed in real-time throughout the whole time period of its execution, and not at the end of the operation. In order to naturally describe these kinds of communications functions, the notion of "stream" was introduced in the ODP reference model (see for instance [Ley 96]).

In this section, we will describe the concepts of a *stream of information* and of a *stream processing function*, such as a communication service, and describe their relevant QoS characteristics.

3.1. The notion of "stream"

The term "stream" is commonly used to represent a flow of information with real-time properties, such as coded audio or video. Multimedia systems contain a number of stream processing components, each component usually realizing a particular stream processing function, such as video encoding or decoding (performed by some hardware or software component of a computer), or data transmission (performed by some communication network). Each stream processing component has usually at least one stream input and one stream output, which may have different real-time properties; for instance, the throughput at the output side of a decoder will in general be bigger than the throughput at its input side.

A multimedia stream has a number of properties which are related to the multimedia semantics represented by the data. We mention only the following:

- media type (e.g. video, audio, video with audio, etc.)
- coding scheme / standard
- resolution (in the case of a video stream: number of pixels per line and column)
- throughput (in bits per seconds)

In the following, we are mainly interested in the throughput of a stream. In the case of variable bit rate, it is important to distinguish such parameters as average throughput, average variation of throughput, and maximum throughput for a given time interval. We note that the other properties may be described using a type system for streams, as described in [Eli 96].

3.2. QoS characteristics of communication services

When a stream passes through a communication network, the processing performed by the network consists of reconstituting at the output point the same stream of bits received at the input point. An ideal communication service would perform this function without any delay. However, most real communication services are imperfect and can be characterized by the following QoS parameters, to be evaluated over a certain time interval:

- Transit delay: the delay between the reception of a segment of the stream at the input point and the reconstitution of the same bits at the output point. As in the case of the throughput, one may distinguish the average delay over a given time interval, as well as the maximum and minimum values.
- Transit delay jitter: the variation of the transit delay.
- Loss rate: ratio between the number of bits lost (not delivered as output) over the number of bits received as input. In practice, since the information is usually transmitted in blocks such as cells, packets or frames, one usually talks about cell loss rate, packet loss rate or frame loss rate.

We note that the throughput is a property of the stream which is transmitted by the network (see Section 3.1), and not a property of the network.

3.3. Specifying QoS characteristics: Requirements and guarantees

The communication requirements for a multimedia stream can usually be characterized by the values of the required throughput (or in the case of variable bit rate: required average throughput and required maximum throughput available over time intervals of specified length) and the maximum delay, jitter and loss rate.

The QoS provided by some network may be characterized by the same parameters, however, the degree of guarantee for providing the characterized transmission service must also be specified. The following degrees of guarantee may be distinguished [Dan 92]:

- deterministic guarantee (which means that the communication service is equal or better than the specified QoS parameters)

- statistical guarantee (which means deterministic guarantee for a certain fraction, e.g. 95%, of the transmitted data blocks, or for a certain fraction of the connections that are established over a long time period)

- target objectives (which means that the networks knows the requirements and tries to satisfy them without providing any guarantee)

- best effort (which means that the network will do as good as it can without considering the particular user requirements); past experience may provide some information about how well the network usually performs

- no guarantee (which means that no prediction can be made; it is practically the same as "best effort")

The Internet normally operates in the "best effort" mode. ATM networks are being designed to provide "best effort" as well as higher levels of guarantee specifically suited for constant and variable bit rate traffic for audio and video streams.

3.4. Concatenation properties

In the context of network interconnection, the resulting end-to-end communication service can be considered to be the concatenation of the communication services of the individual networks through which the information flows. This applies not only to the logical properties of the communication service [Boc 90], but also to the performance aspects [Flo 94]. Assuming that the end-to-end communication service is obtained by the concatenation of n networks, as shown in Figure 1, the QoS parameters P^{ee} of the end-to-end service may be calculated from the QoS parameters P^{i} of the i-th network (i = 1, ..., n) as follows:

$Available\_Throughput^{ee}$ = minimum (for all i = 1, ..., n) of $Available\_Throughput^{i}$

$Delay^{ee}$ = sum (for all i = 1, ..., n) of $Delay^{i}$

$Jitter^{ee}$ = sum (for all i = 1, ..., n) of $Jitter^{i}$ [assuming that the jitter is defined as the difference between maximum and minimum delay]

Jitter^{ee} = square-root of sum (for all i = 1, ..., n) of square of Jitter^{i} [assuming that the jitter is defined as the average deviation of the delay from the average delay, and the delay is assumed to have a normal distribution]

log (1 - Lossrate^{ee}) = sum (for all i = 1, ..., n) of log (1 - Lossrate^{i})

These simple formulae can be used to determine the QoS parameters of the end-to-end communication service of any linear interconnection of communication networks. Moreover, the same formulae hold also for any linear composition of stream processing components that perform other kinds of stream processing functions, as discussed in Section 3.6.

For example, the end-to-end delay for presenting a MM document, which is stored in a file system, is the sum of the delays introduced by the different system components. In the case of the example shown in Figure 1, this sum includes the time required to read a data block, the processing time for the communication protocols in the server station, the delays in the two interconnected networks, the protocol processing delay in the client workstation, and the decoding and presentation delay in the workstation. Similarly, losses may be introduced by the different components. For instance a software decoder may drop audio or video frames when its processing speed is not fast enough, usually due to lack of available CPU power.

It is important to note that the degree of guarantee that can be obtained on the end-to-end basis is the lowest common denominator of the guarantees that can be obtained for each of the components involved. For instance, if one of the networks involved in Figure 1 only provides best-effort service, no end-to-end statistical guarantee can be obtained, unless the performance of the best-effort network is monitored and an alternative system configuration is available which can be used when the performance of the best-effort network does satisfy its target objectives.

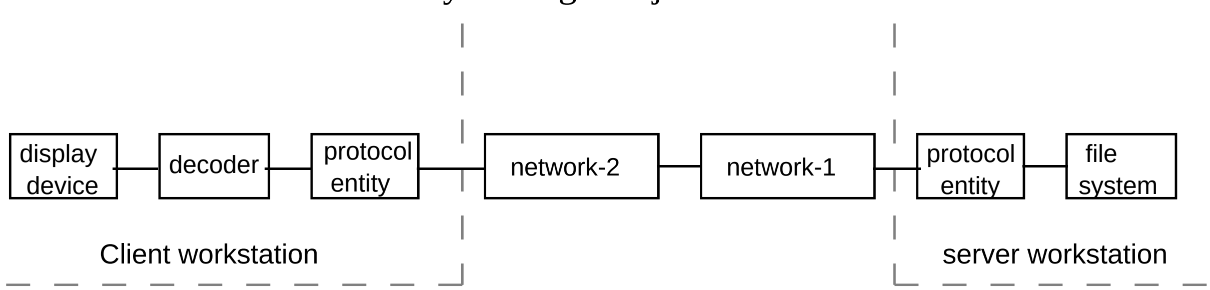

Figure 1. Example of a MM Stream

3.5. Protocol layering

The principle of protocol layering is well-known through the seven layers defined by the OSI Reference Model. Various standardization groups have developed the definition of QoS parameters associated with the communication services [Iso 94, Hol 96] provided by different protocols layers, in particular for ATM [Cci 92], and the transport layer [Dan 92, Met 92] .

During the last years, much attention was given to the question of how the QoS characteristics of one layer, say layer n, depend on the QoS characteristics of the underlying layer and the protocol used in layer n. The answer to this question is particularly complex if the protocol of layer n provides multiplexing.

In practice, however, the multimedia real-time data traffic does not require complex protocols, since occasional data loss is acceptable. Besides acceptable throughput, delay and loss parameters, the real-time traffic for multimedia applications only requires a basic multiplexing function which allows the distinction of different streams. This multiplexing function can be provided by ATM paths and channels or the Internet socket numbers.

Using a minimal protocol suite on top of ATM or Internet IP, the only impact, on the available QoS, of the protocol suite over the network layer would be the following:

(1) A slightly reduced effective throughput due to the header overhead of the protocols.

(2) A recalibration of the loss rate in terms of the application frame loss rate. For instance, in the case of a video stream, a coded video frame may be transmitted as a sequence of m packets. Assuming that a complete video frame is dropped when one of its packets gets lost, then the effective frame loss rate can be approximated, within certain bounds, by m times the packet loss rate. Similar considerations hold for the case of cell losses in ATM networks.

(3) Local processing delay, which adds to the effective delay parameter.

Points (1) and (2) are easily evaluated, while point (3) must be addressed by building an efficient implementation of the protocol suite.

3.6. Other stream processing functions

A distributed multimedia application includes, in addition to the communication networks, various other stream processing components, such as encoders, decoders, audio and video bridges, multimedia file servers and audio / video display equipment. The QoS parameters of these components may be characterized by the same parameters described for communication services in Section 3.2. The definition of processing delay is evident. The definition of a non-zero loss rate is relevant for instance in the case of software decoders that cannot match the speed of the incoming video stream and therefore drop a certain fraction of the video frames.

3.7. More complex configurations

So far, we have only considered the simplest kinds of stream processing configurations, namely linear ones. These configurations are sufficient to describe presentational applications, such as access to a multimedia database, and point-to-point conferencing applications (separate streams in the two directions of transfer).

In the case that different media come from separate sources (for instance in the case of a video clip with a separately stored voice track), the separate monomedia streams must be synchronized at the destination. This requires synchronization protocols between the destination and the different sources and this introduces more complex configurations [Lam 94]. Figure 2 shows a configuration which describes the delivery of a MM document which consists of a video file and the corresponding audio file, stored at different locations. Each media stream has a linear organization, as discussed above, and the synchronization between the streams is controlled in the client workstation involving a control protocol with each server for scheduling the transfer of the respective stream according to the expected delay from the server to the workstation.

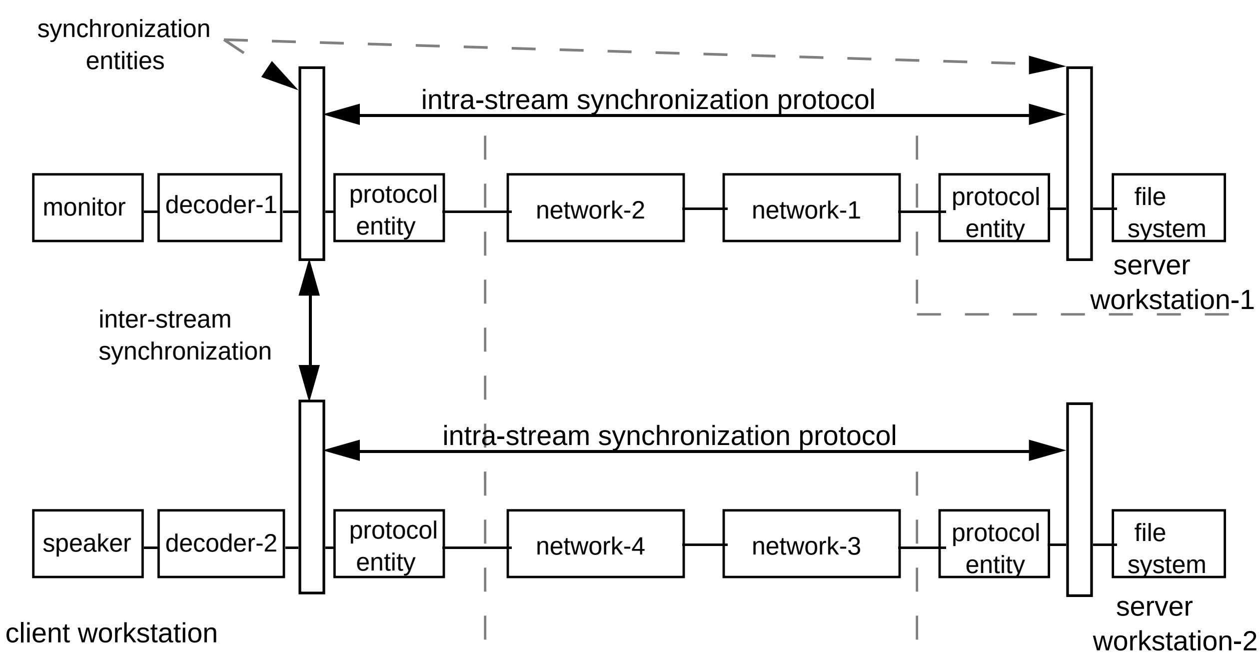

Figure 2. Example of a configuration with media synchronization

In the case of shared applications or multi-party conferencing, the configuration of stream processing become much more complex, including multi-point delivery and intermediate processing functions, such as bridges. We will not further discuss such complex configurations in this paper.

4. Possibilities for QoS adaptation

In this section, we shortly discuss different ways how applications and users may adapt to the given QoS limitations of the underlying system. We first discuss, how one may adapt to unsatisfactory service parameters of a communication network; then we discuss adaptation scenarios involving alternative system architectures and document structures. In the case that the underlying system provides deterministic or statistical QoS guarantees, these adaptations may be planned at the beginning of an session, based on the guarantees obtained during QoS negotiation. In the case of best-effort services, similar adaptations may be desirable to be performed during the execution of an application.

4.1. Adapting to limited quality of the communication service

In the case of network congestion, the available communication service may be affected in the following ways:

(1) The network refuses any new connection request: The only adaptation scenarios are the following:

- to delay the application and to try later, or

- to use another network.

(2) The QoS parameters degrade during a session; in particular, the delay and loss rate increase: If any one of these parameters exceeds the acceptable value, there are the scenarios mentioned under point (1). Alternatively, the loss rate may be decreased by using a protocol with redundancy, however, only by increasing the throughput, and possibly also the delay; this may be an interesting approach if at the same time the application may replace some real-time sensitive media, such as voice or video, by a simpler media, such as text. We note that simply reducing the throughput of the stream will not have any noticeable effect on the network congestion, except if the application under consideration is the main user of the network, or some part of it.

In the case that congestion occurs in the user workstation, the situation may be improved by reducing the throughput of the media stream, for instance by reducing the resolution and/or frame rate of a video stream. Similarly, when enough resources become again available, the reduced quality may be upgraded. Such changes clearly should involve the source of the data stream, which has to switch between different resolution variants. Several existing systems use such adaptations for a single media stream [Gil 91, Tur 94].

The delay parameter is very critical for conversational multimedia applications, such as teleconferencing, but for many other applications, for example access to multimedia databases or video on demand, a delay of a few seconds is easily tolerated. In the case of presentational applications, the communication delay may be compensated by transmitting the data stream a few seconds before the time the presentation is scheduled; however, this requires that the presentation schedule be known in advance.

The delay jitter can be eliminated by buffering the received data at the final destination before its use. However, this introduces an additional delay, equal to the amount of jitter that can be compensated, which may be a problem for conversational applications. It also requires additional buffering space. Data blocks arriving with a jitter larger than the bound that can be compensated will usually be lost.

4.2. Adaptation through alternative configurations

As mentioned above, a basic alternative in the case of network congestion is to use another network. This may not be so easy, since most computers are nowadays only connected to a single network. However, in the case of an overloaded server computer, the switch-over to another server may be a quite reasonable alternative. Alternative service providers are more common than alternative networks.

This leads us to consider configuration alternatives. Such an approach was taken in our News-on-Demand prototype where different variants of multimedia documents may reside on different file servers, and where the servers may be connected to the user's workstation through different networks. The selection of an appropriate system configuration is further discussed in Section 6.

4.3. Adaptation through alternative document structures

As mentioned above, network congestion is difficult to cope with for distributed multimedia applications. In the context of wireless communication, in particular, communication bandwidth is relatively limited and, in addition, may be temporarily even further reduced due to fading. One way to adapt to such situations is to drastically change the structure of the application, for instance, to replace some video stream by some corresponding text to be presented.

It is not clear how such adaptations can be done in a generic manner. In the document structure of our News-on-Demand prototype, for instance, we have planned so-called alternatives, which are monomedia components that may be used as an alternative for a given monomedia component of a different type. For example, the definition of a multimedia document D1 may specify that the video clip V1 could be replaced in the context of this document by the text component T1, which is normally not part of this document [Boc 96]. More examples of adaptation are given in [Muh 95] and [Gec 96].

5. QoS negotiation with the user

5.1. The user's view of service quality

The user should be able to express the desired QoS depending on his needs and his financial capacity. The user's preferences are described in terms of (1) QoS setting for video, audio, still images and text, (2) the cost the user is willing to pay for a given quality, and (3) certain time constraints, such as the maximum acceptable delay for the presentation of a news clip. These preferences must be described in terms of a set of user-perceived characteristics of the performance of the service. It should be expressed in user-understandable language and manifests itself as a number of parameters, such as the value "CD" for audio quality, or the values "TV" or "HDTV" (for more details, see [Haf 95]).

For resource management purposes, the user preferences must be mapped to the internal technical QoS parameters, which were discussed in Section 3.

To facilitate the definition of the user preferences for a given application, we have included in our News-on-Demand prototype the notion of user profile, which contains a given set of user preferences. To avoid repeating the lengthy QoS parameters setting process, the user should be able to store QoS profiles. Then, while starting a new session he selects the desired profile. Furthermore the user can display examples of varying quality in order to see if the profile is pertinent. A set of QoS parameters is associated with each type of monomedia, namely video, audio, text and image. Furthermore to specify the cost and timing constraints, a number of parameters are required. The profile manager provides a set of predefined user profiles that help the user in setting a new profile. A detailed presentation of the profile manager can be founded in [Haf 96b].

For access to multimedia databases, the user generally does not know in what kind of quality the documents would be available. The user profile may contain, in addition to the normal desired quality values, also values for the least acceptable quality limits. In our News-on-Demand prototype, the user profile consists of two so-called *Multimedia* profiles including the information shown in Figure 3. One *Multimedia* profile represents the desired QoS and the other the worst acceptable values.

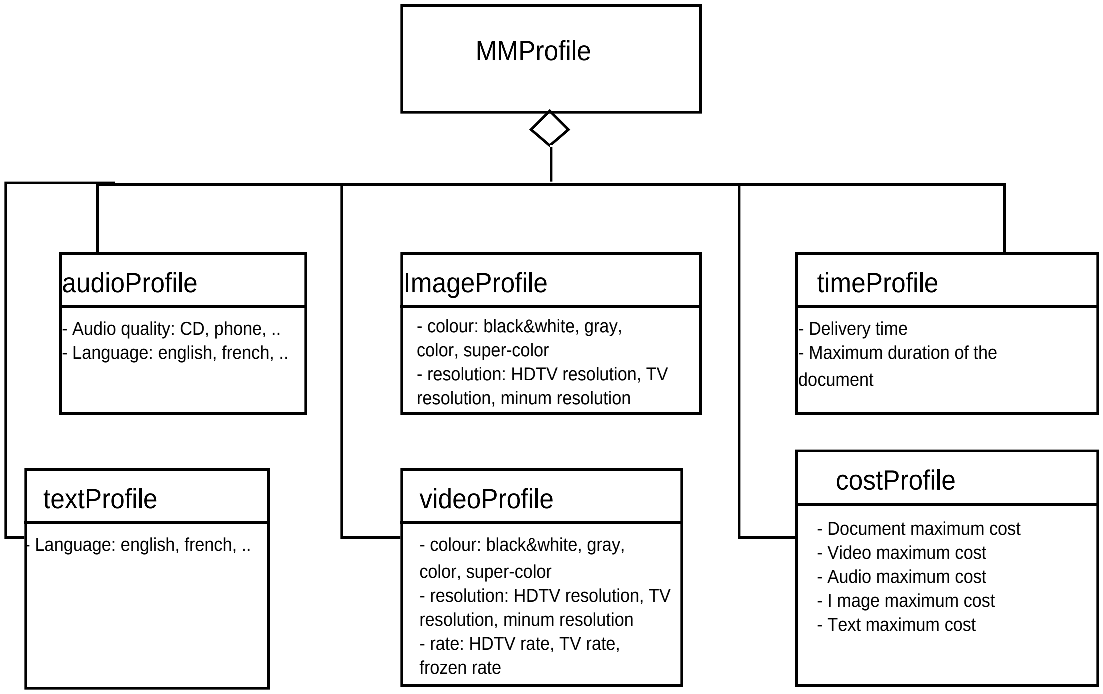

Figure 3. Information structure of a *Multimedia* Profile

5.2. Trade-offs

It is important to note that the selection of the QoS aspects of an application is more difficult if the cost factor is taken into account. It is in general not clear whether a better quality video with higher cost is preferable to a lower quality video variant with lower cost, assuming that the higher cost and the lower quality are within the bounds of the least acceptable quality limits. In order to handle such trade-offs, we have proposed [Haf 96b] that the user should specify the "importance" of each QoS parameter, for instance the importance of video quality and cost. These importance factors can then be used to determine the ordering of the different variants and configurations that may be adopted by the QoS manager.

In the case of video-on-demand and similar applications, another kind of trade-off may be made between presentation quality and the time of the presentation. In this case we assume that the service may be scheduled in the future, for instance, a user may request at 18:00h the viewing of a particular film at 20:00h. When another user requests the same film at 19:00h, the QoS manager may determine that the system is already much loaded and only high-priority requests, which are more expensive, can be granted. However, viewing the film at 20:00h synchronously with the first user could be provided under normal priority and possibly with a cost discount because of the sharing of the file server by the two users. Algorithms for handling such requests with future reservation are described in [Haf 96a].

5.3. User negotiation and adaptation

In the case that the system cannot provide the QoS requested by the user, the system should not simply abort the session, but instead negotiate with the user some mutually agreeable QoS parameters.

In our News-on-Demand prototype, the QoS manager presents in this case to the user the available QoS parameters and asks the user to modify the QoS requirements or to abandon the session. In this way, the user may adapt to the available QoS provided by the system.

6. Configuration management

6.1. A general framework for QoS management

As mentioned above, it is the user-perceived end-to-end QoS characteristics that finally matters, and these characteristics depend on the various system components that participate in the processing of the multimedia streams. Several QoS architectures have been proposed to deal with this global picture (for a review, see [Cam 96]). In addition, the selection of the overall configuration of the application has a primary impact on the available QoS; examples of alternative networks and server computers were discussed in Sections 3 and 5. We think that a key aspect of QoS management is the selection of an appropriate configuration for each instance of an applications.

The framework for QoS management introduced in [Haf 95] proposes the following basic steps for QoS negotiation: defining the user requirements, selecting a functional configuration, and selecting a physical configuration. During the first step, the functional requirements of the application, including multimedia streams, are determined and the QoS requirements of the user are obtained as discussed in Section 5. During the second step, the required functional components of the application configuration are determined. They usually depend on the application in question. An example of a functional configuration for a presentational application is shown in Figure 1. The third step is more complex and starts out with the allocation of the functions to specific physical components. Based on the QoS characteristics available on these components, the functional configuration may be refined, selecting specific coding schemes, for instance. Then there is the important substep of resource allocation, and finally the actual commitment of the resources.

It is clear that the sub-steps of Step 3 involve many opportunities for performance optimization. They must be performed within the QoS constraints provided by the functional characteristics of the application, the user requirements, and the constraints imposed by the QoS characteristics of the physical system components selected. At each of these (sub-) steps, the negotiation process may come to the conclusion that the assumptions made to this point cannot lead to an acceptable physical configuration for the application in question. In this case, the negotiation process should in general backtrack to a previous step and select an alternative proposal, such as an alternative compression scheme, or an alternate physical allocation of certain functions, or an alternate coding format, implying an alternate functional configuration (with a different decoding functional unit) and in the case of the News-on-Demand application, possibly a different server station in which the document variant is stored. In the case that backtracking is necessary up to the first step, we have to introduce the scenario of renegotiation with the user, telling him/her that the requested QoS is not available and proposing possibly several available alternatives from which the user may choose.

We note that the question of resource optimization is in general a very complex problem. An advantage of the above negotiation framework is the fact that it introduces a separation of concern. Up to the allocation of the functional configuration to the physical components a global view predominates, however, the subsequent issues concerning the application of resources within each physical component are assumed to be treated within the context of each component separately. Therefore the many methods developed for the allocation of bandwidth within networks and for the scheduling of real-time processing functions within a single computer system can be used in this context.

Another possibility for the simplification of the overall optimization process is the distinction between static and dynamic management information, and to use the dynamic information only in the last phases of the negotiation process. For instance, we may assume that the first physical allocation selected is based on static performance parameters which are assumed to be available for the different system components, such as the available throughput at the local network access line, tariff tables of the communication network, etc. Subsequently, the dynamic QoS characteristics are checked by requesting the reservation of the resources that correspond to the candidate configuration. If some of the system components, as for instance the network or the file server, cannot provide the requested service (possibly because of temporary congestion) the QoS manager may backtrack to identify another physical configuration.

Our QoS management framework differs in several respects from the earlier QoS management architectures described in the literature [Cam 94, Fer 95, Vol 95, Gop 94, Nah 95, Taw 94]. In particular, these approaches assume that the system components involved in the application are known a priori. Therefore they do not consider any reconfiguration of system components for a given application.

6.2. A QoS negotiation algorithm

We have implemented a QoS negotiation algorithm for distributed MM presentational applications [Haf 96b], which is an instantiation of the above management framework and demonstrates its feasibility.

The algorithm, typically executed within the client workstation, assumes that the workstation is connected to a number of networks, which in turn, lead to a number of server machines, as shown in Figure 4. The objective of the algorithm is to find an optimal system configuration which supports the delivery of the MM document requested by the user with an acceptable, if not desirable, QoS. This means that a server machine and a corresponding network must be selected such that the resulting configuration maximizes the benefit to the user in terms of a trade-off between best presentation quality and lowest cost.

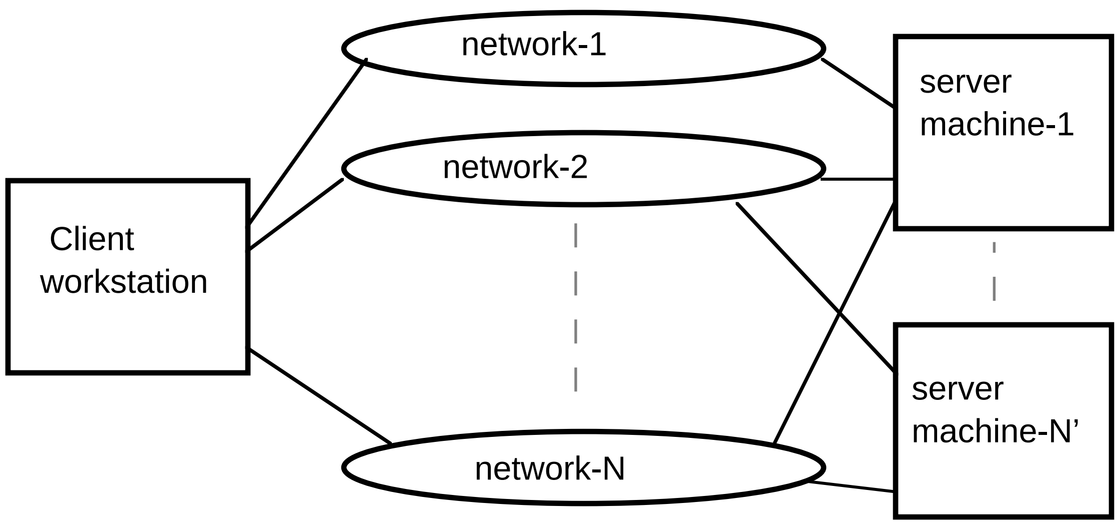

Figure 4. Physical configuration of system for MM presentations

As explained in Section 4.3, it is also assumed that a MM document consists of a number of monomedia components, and that each monomedia may exist in several variants with different locations and QoS characteristics. Our algorithm assumes that, for each monomedia component, the most suitable variant may be selected independently of the other components.

The negotiation procedure for a given monomedia component consists of the following two steps. The first step takes only the static information into account. It produces a list of all variants and intermediate networks which, together with the constraints imposed by the user's workstation, satisfy the static QoS constraints of the user profile. The benefit of each alternative, in terms of the trade-off between best presentation quality and lowest cost, is evaluated, based on the static QoS parameters, and the alternatives are sorted by decreasing benefit. The second step of the negotiation considers the dynamic information. Starting with the best alternative, it checks that the corresponding server and network have the necessary resources to provide the QoS specified by the static parameters. If the resources can not be reserved, the next best alternative will be considered.

If such a reservation has been completed for each monomedia component, the negotiation algorithm completes successfully, and the presentation of the document begins. Otherwise, the user is invited to renegotiate those QoS characteristics of the active profile which could not be satisfied by the most interesting system configurations which were considered by the negotiation algorithm.

This negotiation algorithm is relatively efficient, since it treats each monomedia component separately. In fact, the algorithm has a complexity of M*V*N, where M is the number of monomedia in the document, V is the maximum number of variants of a given monomedia, and N is the maximum number of networks which connect a given monomedia variant to the user's workstation. It is clear that M and N are usually very small numbers.

6.3. Performance prediction and monitoring

The formulae for calculating the end-to-end QoS parameters of a linear chain of stream processing components, described in Section 3.2, can be used for QoS management and prediction. When a user requests a new distributed application, the different stream processing functions required by the application must be allocated to specific hardware components that are available, such as the user's workstation, one or several interconnected networks, possibly a video file server, etc. This allocation, which is part of the QoS negotiation algorithm, should take into account the end-to-end performance requested by the user and the estimated performance of each of the components, in order to assure that the requested performance will be attained.

It is often not easy to predict the performance of each of these components. Operating systems with real-time process scheduling are required to assure some guarantee for the performance of processes that do the stream processing within the environment of the user's workstation, or in a video file server. In the case of "best effort" communication networks, their performance may only be estimated, while present-day ISDN and future ATM networks can provide services with guaranteed QoS parameters that can be selected at connection establishment time.

When the performance of some components cannot be guaranteed, it is useful to foresee on-line monitoring of the effective performance obtained during the execution of the application. Such a monitoring system should know the target QoS parameters and produce a notification to the QoS manager each time that the target value is violated. The QoS manager may then take appropriate actions, such as reconfiguring the system.

Because of the dynamic fluctuations of the QoS actually delivered, certain precautions are usually taken to avoid spurious notifications. The measured QoS value is usually obtained by taking the average over a given measurement interval. Such measurement intervals are usually scheduled at fixed periods. If the measured value obtained during a given interval is below the target value, the measurement may be repeated for N consecutive intervals and only if the measured values in all these intervals was below the target, a notification is sent to the QoS manager. The measurement interval, the period of measurements and the number N are system parameters which may be set by the QoS manager depending on the type of QoS parameter and the desired accuracy, speed and overhead of the measurements.

7. Implementation experience of a News-on-Demand prototype

As mentioned in Section 2, we have implemented a News-on-Demand prototype in a collaboration project involving several Canadian universities. The implementation architecture, shown in Figure 5, follows the high-level architectural design described in Section 2. This system contains software components developed by different groups, including a multimedia database (DBMS) from the University of Alberta [Vit 94], a distributed continuous media file server (CMFS) from the University of British Columbia [Neu 96], a synchronization component from the University of Ottawa [Lam 94], and our QoS management module developed at the Université de Montréal [Haf 95, Haf 96b], and was integrated at the University of Waterloo [Vel 95]. Scalable video encoding was also studied at INRS Telecommunications [Dub 95].

The implementation architecture shown in Figure 5 is strongly influenced by the need for the real-time scheduling of the stream processing functions in the client workstation. Since we had opted for implementation within the Unix environment, we decided to use a real-time thread package which provides deadline scheduling for several threads within a single Unix process. The client software is composed of two parts, the real-time sensitive part consisting of continuous media processing and network monitoring, and the other part consisting of the *user* process which performs user login and database search and the *application* process which performs QoS negotiation and the presentation of the selected document. The remote database has been implemented using ObjectStore [Lam 91], however, the DBMS interface at the client site is a general application programming interface (API) for access to MM databases which can be used with other database systems. We note that Figure 5 does not show the internal structure of the synchronization component nor the interactions of the synchronization component with the CMFS [Neu 96]. More detailed descriptions of the synchronization components and QoS monitor can be found in [Bri 96, Som 95].

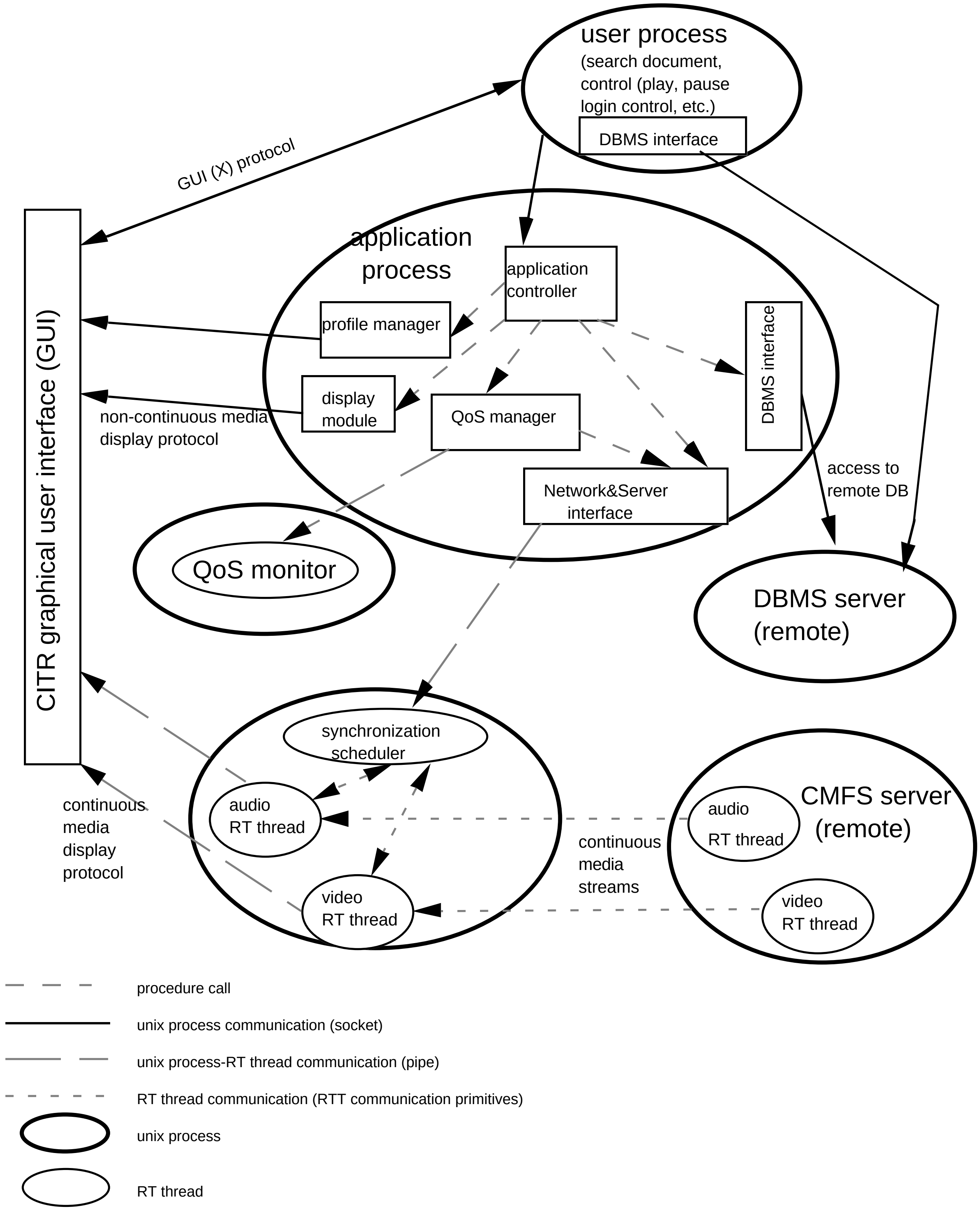

Figure 5. Implementation Architecture of the CITR news-on-Demand Prototype

Immediately after the *application* process is created, the application controller creates the *user* process and enters a loop waiting for primitives, such as play, from the *user* process. The reception of a play primitive results in the following steps:

(1) the application controller gets meta-information on the document to be played via the DBMS interface.

(2) the application controller obtains the current user profile from the profile manager; this may require some interactions with the user via the graphical QoS negotiation user interface.

(3) the application controller initiates the QoS negotiation by sending the negotiation primitive to the QoS manager

(4) the QoS manager executes the QoS negotiation algorithm described in Section 6.2. This algorithm makes use of routines provided by the network and server interface to check the availability of the network and server resources.

(5) the QoS manager returns the results of QoS negotiation to the application controller which may call a routine (provided by the profile manager) to present the results to the user. If the negotiation results are positive, the application will start playing the document.

During the playing of the document, the *user* process enters a loop waiting for user commands, and the *application* process enters a loop waiting for *user* process primitives or notifications from the QoS monitor.

The utilization of an Ethernet and a local ATM switch allowed us to experiment with QoS negotiation and adaptation. For example, in the case that the initially selected video variant was transmitted over the Ethernet, we used a program [Som 96] to generate network congestion. This congestion was detected by the monitoring system and the QoS manager was then able to reconfigure the system to obtain the video from another server over the ATM switch. We performed also some measurements to determine the impact of packet losses on the subjective quality of video and audio presentations; this allowed us to determine the critical values of the loss rate for which a notification should be sent to the QoS manager for renegotiation [Haf 96c]

One of the main difficulties during the implementation of our prototype was lack of compatibility between the different system components developed by the different project partners. Although the different components were designed according specified interfaces which were developed beforehand [Vel 95], different assumptions and interpretation concerning these interfaces resulted in code which did initially not interact correctly. This difficulty led us to define an abstract architecture with abstract interfaces [Boc 96]. These abstract interfaces were the basis for developing the APIs of the QoS and the user profile managers, as well as for the database interface. It was also useful for understanding the prototype and its possible evolution.

8. Conclusions

We presented a few principles applying to QoS management for distributed MM applications. More specifically (1) we identified the characteristics and properties which might be used for stream QoS management, (2) we described various approaches for adaptation of multimedia applications in response to QoS degradations, (3) we presented the issues and some solutions related to QoS negotiation with the human user, and (4) we described a framework for QoS management and the consideration of alternative system configurations. We also gave a short description of a news-on-demand prototype which implements (and thus demonstrates the practicability) of some of the principles described in the paper.

We believe that the principles described here can guide the development of QoS management functions for most of the emerging distributed MM applications. We think that the QoS concerns can be addressed by the following three areas: (1) provisioning of QoS guarantees over virtual network/transport connections, such as planned for ATM networks, (2) resource allocation and real-time scheduling of stream processing functions within the operating system of the user workstation and the continuous media file servers, and (3) overall QoS management at the OSI application level. We tried to show in this paper that the issues in these three areas can be addressed relatively independently from one another, and we have discussed principles for global QoS management concerning the issues of area (3). The issues of areas (1) and (2) are outside the scope of this paper.

Performance models are useful tools for predicting the end-to-end QoS which will be obtained based on the contributions of the different system components involved in providing the service. While the prediction of the performance of each component may be based on more or less complex models specific to each component (e.g. queuing models for a network), the evaluation of the end-to-end performance, based on the performances of the individual components, follows simple formulas (see Section 3) which are based on the stream processing paradigm. On the other hand, the protocols used for QoS management, which are at a meta-level, do not have any real-time constraints, except that they should be fast enough to respond promptly to changes of the system status.

Finally we argue that applications should be able to adapt to varying QoS characteristics that may be available from the underlying systems. In the context of best-effort networks, such as the present Internet, no QoS guarantees are provided. Even if such guarantees will be provided in future networks, it can be expected that the guarantees that can be made for a given application may vary significantly depending on the physical context and time of operation, especially in the context of wireless communications. Therefore it is essential that future multimedia applications be able to adapt to various QoS situations and provide the user with acceptable service quality even in the context of limited bandwidth and temporary fading.

Acknowledgments

We thank the CITR research team for many discussions of the issues discussed in this paper, and we thank Jan Gecsei also for a critical review of a early draft of this paper. An earlier version was presented at the IWQoS'96, Paris, March 1996. 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.

References

[Boc 90] G.v.Bochmann and P.Mondain-Monval, Design Principles for Communication Gateways, IEEE Tr. on Selected Areas in Communications, Vol.8, 1 (Jan. publications), Information Transfer, 1991

[Boc 96] G.v.Bochmann, B.Kerhervé, A.Hafid, P.Dini and A.Pons, Architectural Design of Adaptive Distributed Multimedia Systems, in IEEE International Workshop in Distributed Multimedia Systems Design, Berlin, Germany

[Cam 94] A. Campbell, G.Coulson and D.Hutchison, A Quality of Service Architecture, ACM Computer Communication Review, April 1994

[Cam 96] A. Campbell, Are QoS architectures necessary to meet end-to-end performance guarantees, invited paper at IWQoS'96, Paris, March 1996, (Proceedings published by GMD).

[Cci 92] B-ISDN ATM Layer Cell Transfer Performance, CCITT Draft Recommendation I.35B, 1992

[Dan 92] A.Danthine, OSI95: High Performance Protocol with Multimedia Support on HSLANs and B-ISDN, the Third Joint European Networking Conference, Insbruck, 1992

[Dub 95] E.Dubois, N.Baaziz and M.Matta, Impact of Scan Conversion Methods on the Performance of Scalable Video Coding, SPIE95

[Eli 96] F. Eliassen and J. R. Nicol, A flexible type checking model for stream interface binding, Proc. Intern. Workshop on Multimedia Software Development (MMSD096), Berlin, March 25/26 1996.

[Fer 95] D.Ferrari, The Tenet Experience and the Design of Protocols for Integrated Services Internetworks, Multimedia Systems Journal, November 1995

[Flo 94] P.Florissi and Y.Yemini, Managing Quality of Service in QuAl, in Proceedings of the First International Workshop on Quality of Service, Montreal, Canada, 1994

[Gec 96] J.Gecsei, Adaptation in Distributed MM Systems, Submitted to IEEE Multimedia

[Gil 91] M.Gilge and R.Gussela, Motion Video Coding for Packet Switching Networks-An Integrated Approach, In Proceedings of IT&S/Spie Visual Communication and Image Processing, 1991

[Gop 94] G.Gopalakrishna and G.Parulkar, Efficient Quality of Service in Multimedia Computer Operating Systems, Department of Computer Science, Washington University, Report WUCS-TM-94-04, 1994

[Haf 95] A.Hafid, G.v.Bochmann, B.Kerherve, R.Dssouli and J.Gecsei, On Quality of Service Negotiation in Distributed Multimedia Applications, Technical Report #977, Université de Montréal, Montréal, Canada, 1995 (Submitted to IEEE Journal of Selected Areas in Communications)

[Haf 96a] A.Hafid, G.v.Bochmann and R.Dssouli, A Negotiation Approach with Future Reservations (NAFUR), In Proceedings of the Fourth IFIP International Workshop on Quality of Service, Paris, France, 1996

[Haf 96b] A.Hafid and G.v.Bochmann, A Negotiation Protocol for Distributed Multimedia Presentational Applications, In Proceedings of IEEE HPDC-5, Syracuse, NY, 1996.

[Haf 96c] A.Hafid and G.v.Bochmann, Quality of Service Negotiation in News-on-Demand Systems: An Implementation, Submitted to IEEE PROMS, Madrid, Spain, 1996

[Hol 96] J. Homes et al., Quality of Service - Standards Update, invited paper at IWQoS'96, Paris, March 1996, (Proceedings published by GMD).

[Iso 94] ISO-QoS, Quality of Service Basic Framework-Outline, ISO-IEC JTC1-SC21-WG1 N1145, International Standards Organization, UK, 1994

[Lam 91] C.Lamb, G.Landis, J.Orenstein, and D.Weinreb, The Objectstore Database System , Communications of the ACM, October 1991

[Lam 94] L.Lamont and N.Georganas, Synchronization Architecture and Protocols for a Multi media News Service Application, IEEE International Conf. on Multimedia Computing and Systems, 1994.

[Ley 96] P. Leydekkers and V. Gay, ODP view on QoS for open distributed multimedia environments, paper at IWQoS'96, Paris, March 1996, (Proceedings published by GMD).

[Met 92] B.Metzler, I.Miloucheva and K.Rebensburg, Multimedia Communication Platform: Specification of the Broadband Transport Protocol XTPX, CIO, Race Project 2060, 60-TUB-CIO-DS-A-002-b2, 1992

[Muh 95] M.Muhlhauser and J.Gecsei, Services, Frameworks, and Paradigms for Distributed Multimedia Applications, Submittedto IEEE Multimedia, 1995

[Nah 95] K.Nahrstedt, An Architecture for End-to-End Quality of Service Provision and its Experimental Validation, Ph.D. Thesis, University of Pennsylvania, 1995

[Neu 96] G.Neufeld, D.Makaroff and N.Hutchison, Design of a Variable Bit Rate Continuous Media File Server for an ATM Network, SPIE 96, 1996

[Ray 93] K. Raymond, The reference model of Open Distributed Processing: A tutorial, Proc. of the 1st International Conference on ODP, Berlin, J. de Meer, B. Mahr and O. Spaniol Eds., pp.3-14.

[Som 96] R.Somaling, Network Performance Monitoring for Multimedia Networks, Master Thesis, School of Computer Science, McGill University, Montreal, Canada, 1996

[Taw 94] W.Tawbi and E.Horlait, Expression and Management of QoS in Multimedia Communication Systems, Annals of Telecommunications, June 1994

[Tur 94] T.Turelli, The INRIA Videoconferencing System (IVS), Connexions-The Interoperability Report, Vol. 8, No 10, 1994

[VEL95] R. Velthuys (Ed) (1995). CITR Broadband Services: the March'95 demo; Technical Report, University of Waterloo, March 1995

[Vog 95] A. Vogel, B. Kerhervé, G. Bochmann, J. Gecsei (1995). Quality of Service Management: a survey. IEEE Journal of Multimedia Systems,Vol 2 no 2, summer 1995

[Vol 95] C.Volg, L.Wolf, R.Herrtwich and H.Wittig, HeiRat -Quality of Service Management for Distributed Multimedia Systems, Multimedia Systems Journal, November 1995

[VIT94] C.Vittal, T.Ozsu, D.Szafron and G.El-Medani, The Logical Design of a Multi Database for a News-on-Demand Application, Technical Report 94-16, University of Alberta, Alberta, Canada, 1994