Overload control in distributed multimedia systems

Bechir Alaya, Claude Duvallet, Bruno Sadeg

LITIS, UFR des Sciences et Techniques
25 rue Philippe Lebon, BP 540
F-76058 LE HAVRE CEDEX
Firstname.Lastname@litislab.fr

RÉSUMÉ. *Les applications multimédia gèrent des volumes importants de données. L'exploitation de ces données doit se faire en respectant les contraintes temporelles permettant de lire les paquets vidéo avec une certaine fluidité. Lorsque les contraintes temporelles ne sont pas respectées, la qualité de service fournie aux utilisateurs diminue. Partant du constat qu'il existe des similarités entre les applications multimédia et les SGBD temps réel, notre approche consiste à exploiter les travaux concernant la gestion de la qualité de service dans les SGBD temps réel afin de les appliquer aux systèmes multimédia distribués. Dans cet article, nous proposons une nouvelle architecture permettant la gestion efficace des données multimédia et d'améliorer la qualité de service fournie aux clients lors des variations importantes de la charge d'utilisation du système. Notre technique consiste à faire varier la qualité des flux vidéo transmis sur le réseau en utilisant les particularités du standard MPEG et en appliquant la notion de contraintes (m,k)-firm.*

ABSTRACT. *The multimedia applications manage voluminous quantities of data. Its exploitation must respect the temporal constraints permitting to read the video packets with a certain fluidity. When the temporal constraints are not met, the quality of service (QoS) provided to users decreases. Considering there are similarities between the multimedia applications and the Real-Time Databases Systems (RTDBSs), our approach consists to exploit the works concerning the management of the QoS for the multimedia system. In this paper, we exploit some results obtained in QoS management in RTDBSs, and we apply them to multimedia applications, because of similarities existing between these two fields. We propose a new method allowing to have an efficient management of data and to control the QoS provided to clients according to the system congestion.*

MOTS-CLÉS : *Systèmes multimedia distribués, Boucle de rétroaction, Qualité de service, Format MPEG, Contraintes (m,k)-firm*

KEYWORDS: *Distributed multimedia systems, Feedback control loop, Quality of service, MPEG format, (m,k)-firm constraints.*

1. INTRODUCTION

The progress in the networks domain and the desirable improvement allow to exploit of new services such as those related to multimedia applications. These recent years, there are many researches which are not only interested minimizing computation time, but also to satisfy application time constraints, i.e. deadlines, release times, etc. In this paper, we focus on a kind of these applications : multimedia applications. These applications must exchange very important quantities of data and their treatments require to be done before fixed dates to guarantee an acceptable quality of service (QoS) in the streams presented to the users. RTDBSs are the systems adapted to such data management while dealing with a certain QoS [RAM 93] [RAM 04]. In multimedia applications, the management of the QoS of the video packets allows to answer to these new needs. Since about a decade, researchers try to adapt feedback control scheduling for multimedia systems. Adapting efficiently existing techniques to the video packets management without modifying the initial infrastructure is a new challenge. The main issue is the adaptation of the available resources (bandwidth, buffers size, video servers, etc.) and the proposition of news techniques to manage streams when instability periods (overload or underutilization) occur. The goal is to assure an acceptable QoS provided by the system to users, while respecting the multiple requirements of the video streams. Several studies have focused on the definition of mechanisms and strategies which allow the system to provide an acceptable QoS. Many works on QoS management in RTDBSs have been done [AMI 06] [KAN 02a]. Almost all these works are based on a feedback control scheduling architecture (FCSA) that controls the system behavior thanks to a feedback loop. The feedback loop begins to measure the performances of the system in order to detect overload periods. Then, according to the results observed, the values of the parameters are modified to adjust the system load to the real conditions. As these conditions always vary, this process is repeated indefinitely. Because of the similarities existing between RTDBSs and multimedia applications, in this paper, we propose to apply the results obtained on the QoS management in RTDBSs to multimedia applications. The main objective is to allow to design multimedia applications that will be able to provide the QoS guarantees and a certain robustness when user's demands quickly grow up and/or when the network becomes congested. These works are especially applied to video on demand (VoD) applications. The remaining of the paper is organized as follows. Section 2 describes the management of the quality of service in distributed multimedia systems. In Section 3, we develop our approach which allows to increase the applications QoS during overload periods (network congestion). Some simulation results are given in Section 4. In Section 5, we conclude the paper and give some perspectives.

2. QUALITY OF SERVICE IN DISTRIBUTED MULTIMEDIA SYSTEMS

Our approach consists in taking into account researches already done on the management of QoS in RTDBSs [KAN 02b] [AMI 03] and their adaptation to multimedia systems. QoS in a multimedia application may be defined as the requirements in terms

of bandwidth, quality of visualization, delay and rate of video packets loss. To this purpose, we propose an adaptation of a method based on feedback control architecture to distributed multimedia systems [DUL 04]. We exploit the notion of (m,k)-firm constraints proposed in real-time sytems [BUC 95] and in RTDBSs [HAN 01][AND 96]. This adapted method is called FCA-DMS (Feedback Control Architecture for Distributed Multimedia Systems). We apply a control of the network congestion by discarding or not some multimedia frames of certain types according to the network state, notably to the shared bandwidth. We show that this will increase the QoS provided to the users.

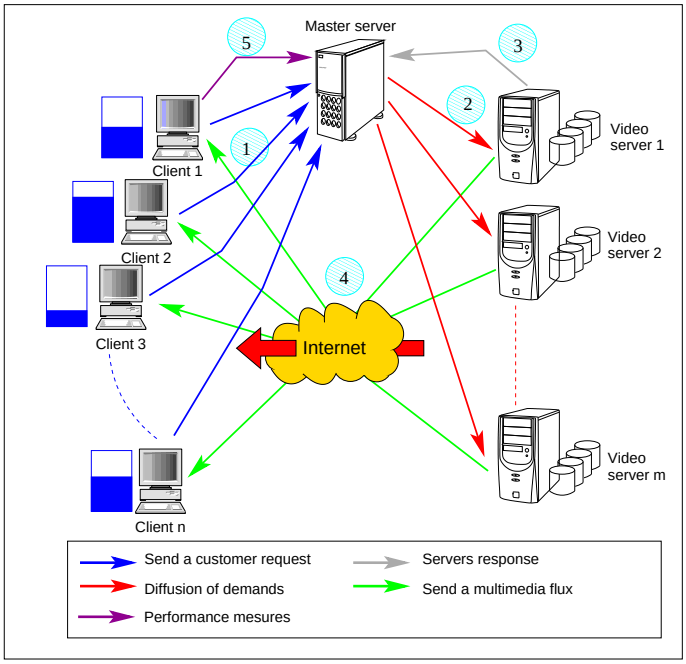

Figure 1. *Functional model of the FCA-DMS architecture.*

2.1. *Feedback control architecture*

In a previous work, Natalia Dulgheru has proposed an architecture, named QM-PEGv2 [DUL 04] which deals with distributed multimedia systems (cf. figure 1). The architecture proposed contains three main components :

– ***Master server :*** it accepts requests from clients, chooses the video servers able to serve the demand, supervises the system state and adjusts the video streams in order to maintain the QoS initially fixed.

– ***Video servers :*** They run under the master server control and send the video packets to the clients.

– ***Clients :*** they send requests to the master server and receive the video frames from the video server. When a state change occurs, they send a feedback report to the master server.

In the following, we describe briefly a typical procedure which is executed when a video on demand is requested, based on FCA-DMS architecture :

1) A client sends a request to the master server to get a video, with a certain level of QoS.

2) The master server broadcasts the request to the video servers available in the system.

3) The video servers send back their response to the master server, which chooses one among them.

4) A stream is opened between the chosen video server and the concerned client.

5) The master server asks the video servers to adapt their QoS, when necessary.

The feedback loop consists on adapting the QoS according to the load system conditions (servers and network congestion). The system observes the QoS obtained by the client and, if necessary, asks the concerned video server to improve it.

2.2. *Adaptation of QoS and feedback control loop*

In order to improve the QoS, the system increases or decreases the number of transmitted frames of certain types. To this purpose, we based our action on the characteristics of the standard MPEG format [ISO 94], that defines a mechanism to code frames at the time of the video compression. A video sequence enters the system. It's then compressed and coded according to three types of frames : *Intra frames* (I), *Predicted frames* (P) and *Bidirectional frames* (B). I frames are references frames. P frames allow to rebuild a frame using an I frame. B frames use I frames and P frames to rebuild a sequence. Therefore, I frames are the most critical. To decrease the eventual network congestion, it is necessary to remove some frames from a video sequence, but these suppressions must be done in a controlled manner. We propose in the following section a method based on the controlled frames suppression in order to control the QoS provided to users. Using the feedback loop allows to stabilize the system during the instability periods [BOU 05]. It is based on the two principles : observation and auto-adaptation.

The observation principle consists of observing the results obtained by the system and checking if the current QoS observed is consistent with the QoS initially required, e.g. in VoD application, the system checks if the videos sequences are presented to users without interruptions. The auto-adaptation consists for the system to adapt the results according to the QoS needed par the clients, by adjusting some network and video

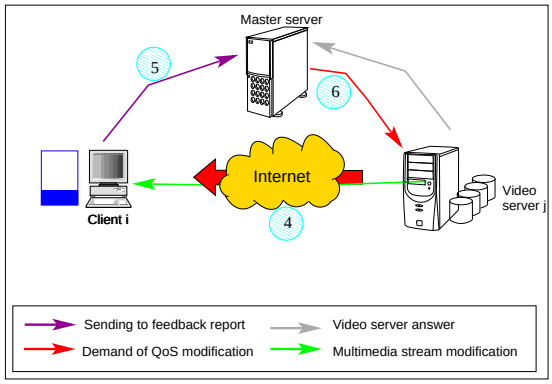

Figure 2. *Adapted feedback loop for multimedia applications.*

parameters, e.g. the system increases or decreases the number of accepted frames[1]. In this way, the feedback control loop ensures the stability of the system.

3. A NEW METHOD TO CONTROL THE SYSTEM CONGESTION

3.1. *Replication strategy : a method to control the video server congestion*

A video server (VS) can distribute only videos stored on its disks. If a video is not accessible on several servers (one VS contain this video), the probability that this last VS will be saturated increases. Therefore, it is necessary to define a new distribution strategy of video packets in order to have another video server which is used to answer to the customer request. The saturated video server sends a request to the nearest video servers. Each video server behaves according to one of the following three scenarios :

1) It possesses the video and it is able to treat the request (it is not saturated).

2) It possesses the video but it is unable to treat the request (it is saturated).

3) It doesn't possess the video, but it is probably able to treat the request because it is not saturated.

In the two first cases, the replication strategy is not established. In the last case, the case manager, that has to control replication, sends an order to the saturated VS to start the replication. Consequently, the case manager elects a VS among those that answered and that are not saturated. The choice of the VS is done in order to get the best possible QoS. The demand returns again to the monitor, in order to allow to terminate the replication, then the monitor restarts works.

1. note that I frames (critical) are not removed

Algorithm 1: Replication strategy

```
master server : MS
client : cl
video server : VS
neighbour VS : CVS
selected CVS : SCVS
begin
    SCVS = Ø
    VS is saturated and has the video
    For all CVSi do
        send-request (VS, CVSi)
        if (CVSi has not the video and not saturated) then
            SCVS = Ø
            exit-for
        else
            if (CVSi has the video and is not saturated) then
                put CVSi in SCVS
            end if
        end if
    end for
    if (SCVS ≠ Ø) then
        choose (CVSj in SCVS)
        video-replication (VS, CVSj)
    end if
end
```

3.2. *The (m,k)-frame method, a technique to control the network congestion*

3.2.1. *(m,k)-firm method*

According to certain conditions, the system load varies from overload state to under-utilization state and vice-versa. Indeed, since the number of video servers sending the video packets is unknown, sometimes this causes severe damages on the service level provided to clients. Consequently, the number of transmitted packets is also unknown and can be important. Moreover, when a high number of video packets access to network resources, it is necessary to keep a high priority level for more critical packets (I frames, then B frames, then P frames)[DUL 04][NG 00].

We propose an approach based on (m,k)-firm method [HAM 95]. The (m,k)-firm model is characterized by two parameters m and k. An application is said under (m,k)-firm real-time constraint if at least m operations among k consecutive operations meet their deadlines. We adapt this method to the context of multimedia applications. A multimedia operation consists of sending/receiving a video frame. To adapt this method, we consider that m video packets among k must be correctly sent. To this purpose, we propose a new technique of video packets management crossing the network, called (m,k)-frame.

A video stream is decomposed into several classes according to their tolerance to the loss of frames characteristics, i.e. each class contains the video packets of similar (m,k)-frames constraints. The three classes we consider, refer to the three types of frames : I, B and P. With this technique we realize a trade off between the shared resources and the QoS granularity in the same class of a video stream.

3.2.2. *Quality of service adaptation*

In this work, we focus on the adaptation of the video stream to the network state. We assume that measures of the network capacity are available, in one hand, and that we have an important number of frames to send, on the other hand.

The three classes of frames (I, B and P) are used to adapt the quality of stream sent to the network. We consider the following constraints : (m_I, k_I)-frame, (m_P, k_P)-frame and (m_B, k_B)-frame, i.e, m_i frames of a certain type must be received among the k_i frames sent. Then the network capacity is measured by the formula : $m_I + m_P + m_B$. Recall that I frames are the most critical. The parameters are ordered in the following manner : $m_I > m_P > m_B$. We usually have $m_I = k_I$, i.e., I frames are critical and it is forbidden to remove them.

We assume the situation where the network, whose current capacity is N, is congested. We also assume that QoS_{max} is the quality of the stream to send. To be consistent with the network capacity, it is necessary to remove ($QoS_{max} - N$) frames. Therefore, we have to degrade the quality of the MPEG stream. When we apply no method of congestion control, frames will be randomly removed, i.e. they are lost by the network, causing the degradation of the video presentation, notably if some I frames are removed. Here, we apply our (m,k)-frame method, which consists of removing frames in an intelligent manner. We have : (1) QoS_{max}=$k_I + k_P + k_B$, and (2) N=$m_I + m_P + m_B$. The number of frames to remove is then : $QoS_{max} - N = (k_I - m_I) + (k_P - m_P) + (k_B - m_B)$, where $k_I = m_I$ (the most critical frames are removed).

3.2.3. *Bandwidth fair sharing*

With the previous assumptions, we deal with the broblem of sharing bandwidth between servers in case of network congestion. In the previous section, we have seen how to reduce the QoS at the stream level, according to the available capacity of the network. Here, we need to share fairly the bandwidth between all sources that wish to send a stream. We compute the total capacity needed by all servers. Then, we compute R, the ratio between the needed capacity and the available network capacity.

$$R = \frac{N}{(RequiredCapacity)}$$

Example : let 3 video servers wishing to send flows of 40, 30 and 20 frames per second respectively. The total capacity of the network needed to answer to this demand must be 40+30+20=90. If, however, the network only arranges a capacity of 75 frames

per second, it is not able to sent all the frames. The ratio R is computed as follows : (75/90)*100 = 83.33%. Then, we apply this rate to each of the three required capacities 40*83.33%=33, 30*83.33%=25 and 20*83.33%=17. If we sum the three obtained numbers, we find 75 frames per second. This corresponds to the actual capacity of the network.

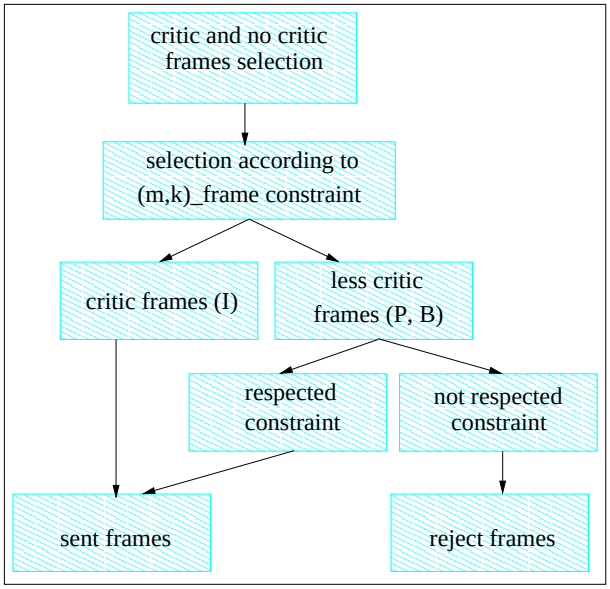

Figure 3. *The (m,k)-frame algorithm.*

In the following, are listed some advantages of the bandwidth fair sharing :

– To control the congestion of the network by fair sharing resources between all streams. A bad stream doesn't affect the service provided to the other streams. Only this service will be concerned if a stream wants to consume more resources than available.

– To guarantee an acceptable bandwidth and delay.

– To guarantee a link sharing between the different classes of service.

4. SIMULATIONS

We have studied and evaluated the behaviour of system and its adaptation to different load variations. The performance evaluation is undertaken by a set of simulation experiments, where various parameters have been varied.

Frames I for 9000 sent frames	Usables Frames	Lost Frames	Received Frames	Sent Frames
Before (m,k)−frame selection	67,73	32,27	67,73	100.00
After (m,k)−frame selection	100,00	0.00	100,00	100,00

(a) For I frames.

Frames P for 9000 sent frames	Usables Frames	Lost Frames	Received Frames	Sent Frames
Before (m,k)−frame selection	44,22	34.27	65.73	100.00
After (m,k)−frame selection	60.00	40.00	60.00	60.00

(b) For P frames.

Frames B for 9000 sent frames	Usables Frames	Lost Frames	Received Frames	Sent Frames
Before (m,k)−frame selection	32,98	33,12	66,88	100.00
After (m,k)−frame selection	45,90	54,10	45,90	45,90

(c) For B frames.

Frames I, P and B for 9000 sent frames	Usables Frames	Lost Frames	Received Frames	Sent Frames
Before (m,k)−frame selection	38,69	33.33	66.67	100.00
After (m,k)−frame selection	53,93	46.07	53.93	53,93

(d) For I, P and B frames.

Figure 4. *Simulation parameters.*

4.1. *Presentation of the simulator*

Our simulator is based on the architecture components presented in Section 2. A master server serves all video servers participating to the video diffusion. Moreover, the master server allows to add new video servers, when necessary. A new server is assigned a number and a list of accessible objects. After starting the master server, the video servers who want to participate in the video distribution, refer themselves to the master server, and then get a number. In order to make a request, a client dialogs with the master server that distributes the request to the available video servers. When the master references a video server able to satisfy the client request, i.e. the server is able to provide the required QoS, it sends the video server reference to the client. Then, the broadcast from the video server can begin. After some time, the client sends to the server the QoS level he obtained. The three parts of the architecture of this simulator are modelled thanks to an object modelling language and realized in JAVA language.

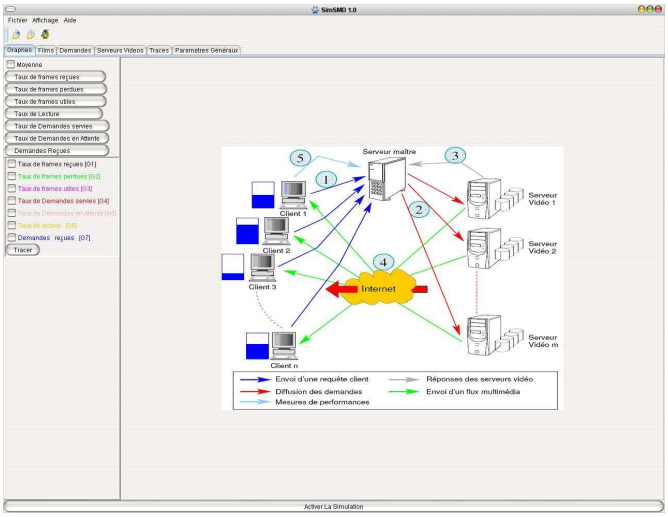

Figure 5. *The simulator architecture*

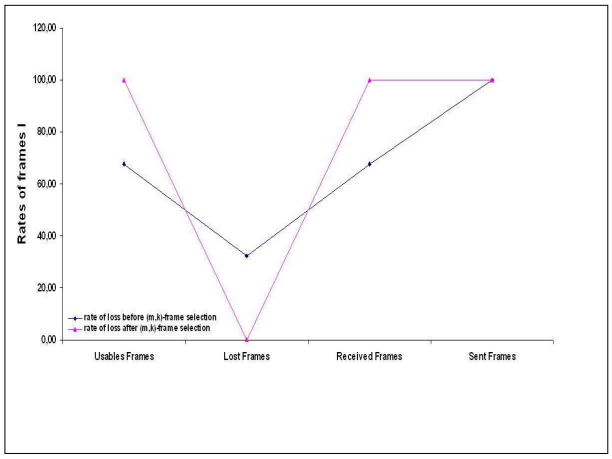

Figure 6. *For I frames.*

4.2. *Simulations objectives*

The objective of the simulations is to demonstrate how our method, called (m,k)-frame, is able to adapt the QoS to the real conditions of a multimedia application, according to the current system load. Notably, the system must adjust the QoS when

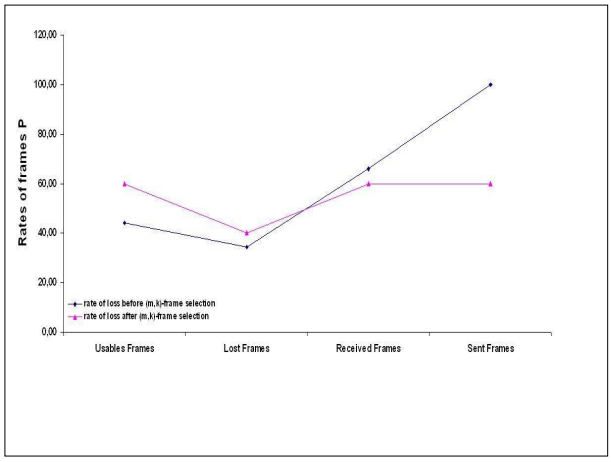

Figure 7. *For P frames.*

the client number that does requests varies (dynamic arrived of clients). The network congestion can have different sources :

– *internal :* when there is a large number of clients doing requests in the system. We can limit this client number by using an admission controller located at the master server level.

– *external :* when the network is used by other applications that can cause the congestion.

Our architecture must adjust the QoS by reducing the number of frames broadcast in the network.

4.3. *Simulation results*

We have tested two possibilities of frames removal in our simulator :

– *Before (m,k)-frame selection :* here, the removal of B frames doesn't depend on P and I frames and the removal of P frames doesn't depend on I frames. The removal of I, P and B frames is randomly done, i.e, the probability to remove a P and I frames is equivalent to that of removing a B frame because they are independent.

– *After (m,k)-frame selection :* in this case, do not remove I frames, because, I frames are critical and it is forbidden the remove them (m_i=k_i). there exist dependencies between P frames and B frames. When a P frame is removed then all B frames that depend on this P frame must be removed because they become useless.

Of course, the results obtained *"after (m,k)-frame"* selection are better than those obtained in *"before (m,k)-frame"* selection.

In Figures 6 and 9, we note that the loss rate of I frames is less important than other types of frames. We particularly distinguish the following cases :

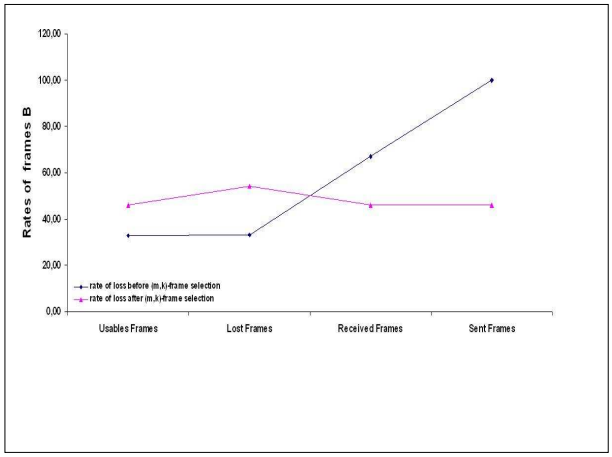

Figure 8. *For B frames.*

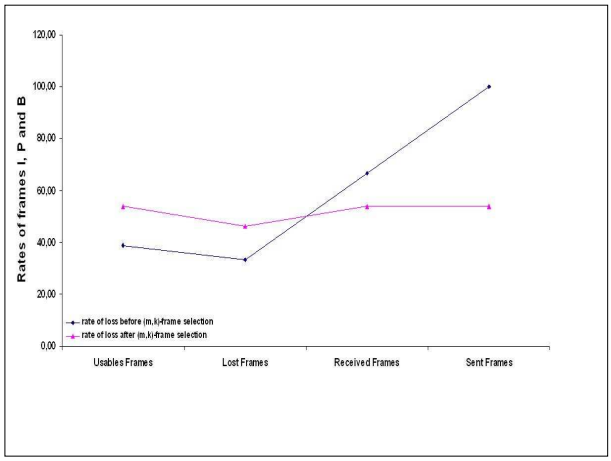

Figure 9. *For IPB frames.*

1) When no frame management method is used : we note that, in Figure 6, only ∼67,73% of I frames sent are received and usable. The other frames are lost. For P frames (Figure 7) and B frames (Figure 8), almost the same rate of frames are received (62%), but the rate of used frames is only about 40%, because P and B might be removed randomly.

In Figure 9, when we consider all types of frames, we obtain approximatively the same results as Figures 7 and 8.

2) When (m,k)-frames method is used : we note, in Figure 6, that all I frames sent are received without any loss and they are all usable. In Figures 7, 8 and 9, we note that only m frames among k are sent. However, we note also that all frames sent, are received and used. Consequently, frames selection is made at the time where the stream enters the system.

Finally, we note, in Figure 9, the rate gap between I, P and B frames with and without (m,k)-frames model : there exists a clear difference between the three types of frames : the selective packets reject is very efficient to maintain an acceptable QoS to the client while minimizing the critical packets loss, i.e. I packets.

In case of network overload, the QoS degradation of the MPEG stream is acceptable with (m,k)-frame, since I frames are more significant than B and P frames for the decoding process, to efficiently rebuild the original video.

5. CONCLUSION AND FUTURE WORKS

While current resource management systems provide mechanisms which provide reliability with respect to QoS, it is not sufficient since there are many well established application scenarios where QoS management is required, e.g., distributed multimedia systems. In this paper, we proposed an improvement of the feedback control architecture for distributed multimedia systems (FCA-DMS). Our objective is to provide a deterministic temporal guarantee according to the temporal constraints of real-time video streams. Our main contribution concerned the adaptation of (m,k)-firm constraints for the video packets and the establishment of a video replication strategy. A possible extension of this work consists in enhancing the architecture presented. Notably, in order to bring some breakdowns tolerance because of the presence of only one master server. We have also highlighted the importance of the (m,k)-frames improvement and have given a certain priority to the QoS modification demand, in order to increase its reliability and robustness and to converge towards the QoS specified by the client.

Simulations results allow us to validate the feasibility of our technique and should allow to provide results demonstrating the real contribution of this new work.

An other possible future work would consist in building a real video on demand server based on the architecture that we propose. We will take into account of frames storage management and frames organization in video servers, notably, the most efficient manner (from QoS point of view) to videos broadcast between the different video servers. This work requires to compare the performances obtained when using different manners to organize and store videos.

6. Bibliographie

[AMI 03] AMIRIJOO M., HANSSON J., SONG S., « Error-Driven QoS Imprecise Management in Imprecise Real-Time Databases », *Proc. of Euromicro Conf. on Real-Time Systems (ECRTS'03)*, Portugal, 2003, p. 63-72.

[AMI 06] AMIRIJOO M., HANSSON J., SON S. H., « Specification and Management of QoS in Real-Time Databases Supporting Imprecise Computations », $IEEE\ Transactions\ on\ Computers$, vol. 55, n° 3, 2006, p. 304-319.

[AND 96] ANDLER S., HANSSON J., ERIKSSON J., MELLIN J., BERNDTSSON M., EFTRING B., « DeeDS : Towards a Distributed and Active Real-Time Database System », $ACM\ SIGMOD\ Record$, vol. 15, n° 1, 1996, p. 38-40.

[BOU 05] BOUAZIZI E., DUVALLET C., SADEG B., « Management of QoS and Data Freshness in RTDBSs using Feedback Control Scheduling and Data Versions », *Proceedings of* 8^{th} *IEEE International Symposium on Object-oriented Real-time distributed Computing (ISORC'2005)*, Seattle, United State, May 18-20 2005, p. 337-340.

[BUC 95] BUCCI G., CAMPANAI M., NESI P., « Tools for specifying real-time systems », *Journal of Real-Time Systems*, vol. 8, 1995, p. 117-172.

[DUL 04] DULGHERU N., « Management of QoS in Distributed MPEG Video », Master's thesis, University of Linkoping, 2004.

[HAM 95] HAMDAOUI M., RAMANATHAN P., « A Dynamic Priority Assignment Technique for Streams with (m, k)-$Firm$ Deadlines », *IEEE Transactions on Computers*, vol. 44, n° 4, 1995, p. 1325-1337.

[HAN 01] HANSSON J., SON S., « Overload Management in RTDBs », *Real-Time Database Systems : Architecture and Techniques*, chapitre 10, p. 125-140, Kluwer Academic Publishers, 2001.

[ISO 94] ISO/IEC 13818-2, « Information Technology-Generic Coding of Moving Pictures and Associated Audio, Part 2 : Video », Recommendation ITU H.262, International Standardization Organization, November 1994.

[KAN 02a] KANG K., SON S., STANKOVIC J., ABDELZAHER T., « A QoS-Sensitive Approach For Timeliness and Freshness Guarantees in Real-Time Databases », *Proceedings of the Euromicro Conference on Real-Time System*, 2002, p. 203-212.

[KAN 02b] KANG K.-D., SON S., STANKOVIC J., « Service Differentiation in Real-Time Main Memory Databases », 5^{th} *IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (IEEE ISORC'02)*, Washington D.C, 2002, p. 119-128.

[NG 00] NG J., LEUNG K., WONG W., « Quality of Service for MPEG Video in Human Perspective », rapport, 2000, Hong Kong Baptist University.

[RAM 93] RAMAMRITHAM K., « Real-time databases », *Journal of Distributed and Parallel Databases*, vol. 1, n° 2, 1993, p. 199–226.

[RAM 04] RAMAMRITHAM K., SON S., DIPIPPO L., « Real-Time Databases and Data Services », *Real-Time Systems*, vol. 28, 2004, p. 179-215, Kluwer Academic Publisher.