A Fibre Channel Dimensioning for a Multimedia System with Deterministic QoS

Laurent George[1], Dana Marinca[2], and Pascale Minet[3]

[1] University of Paris 12, LIIA, 120 rue Paul Armangot, 94400 Vitry sur Seine, France,
george@univ-paris12.fr
[2] University of Versailles, 45 avenue des Etats-Unis, 78035 Versailles Cedex, France
dimarinca@free.fr
[3] INRIA, Rocquencourt, BP105, 78153 Le Chesnay Cedex, France
pascale.minet@inria.fr

Abstract. We propose to use Fibre Channel (FC) technology in multimedia systems offering Video on Demand (VoD) services. The Storage Area Network (SAN) is based on (i) FC-loops connecting magnetic disks and on (ii) FC-switches connecting loops to servers. We show how to dimension FC-loops to offer a deterministic guarantee of Quality of Service to the VoD clients. The performance results of this analysis, confirmed by already published simulation results, enable to determine the optimal number of disks connected to a loop and the maximum number of clients acceptable by a loop. We study the influence of disk performance and determine the best number of blocks to retrieve per disk request.

Introduction

Multimedia systems can be used in many domains: entertainment in hotels, telelearning, production in radio/TV studios,... In this paper, we are concerned with the design of multimedia systems providing VoD (Video on Demand) to their clients. The client may interact by means of VCR commands (i.e. start/stop, pause/play, and jump backward/forward). We are interested in multimedia systems providing a deterministic guarantee of Quality of Service (QoS) to their clients. A multimedia system consists of different components in charge of storage/retrieval of multimedia data, network communication, and system activity control. Each component contributes to ensure the end-to-end QoS. In this paper, we focus on the storage system, a main component of the multimedia system. We propose to use a Storage Area Network (SAN) based on Fibre Channel (FC) technology. Indeed, FC offers a high performance environment for the communications between computers and the storage system and allows a very scalable architecture. We show how to dimension such a system based on a worst case analysis. We determine the maximum number of acceptable clients and the optimal number of disks per loop. We study the influence of disk performance, and size of the data retrieved by the disk. The worst case analysis can be used by the admission control to decide on the acceptance of a new client. If accepted, this client will receive a deterministic QoS guarantee.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 575–587, 2002.

This paper is organized as follows. In section 1, we briefly present the components of a multimedia system and give the properties that must be achieved by such a system. In section 2, we describe the main features of Fibre Channel and a SAN architecture based on this technology. In section 3, we propose a performance analysis of an arbitrated loop connecting servers and magnetic disks. We first recall classical results for real-time non-preemptive uniprocessor scheduling. These results are then applied to disks scheduling and FC-loop access scheduling. This system behavior has been simulated ([12], [13]) and the associated results have been published. These results are used to validate our analysis. Then, we show how to use our results to dimension the storage system. Finally, we conclude.

1 Multimedia Systems

In this section we describe the general architecture of a multimedia system, defining the required properties to achieve the requested QoS.

1.1 General Architecture of a Multimedia System

A multimedia system consists of four main components: the servers, the storage system, the network and the clients. A server is in charge of transmitting multimedia contents from the storage system to the clients or from a multimedia source to the storage system. In this paper we assume that the servers access the storage system by means of a network, as illustrated by figure 1.

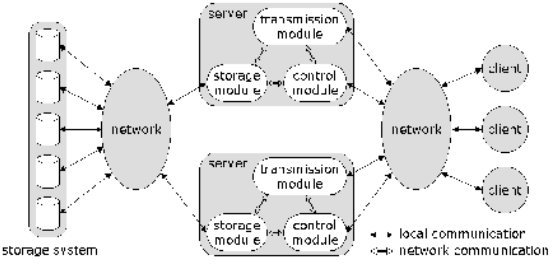

Fig. 1. General multimedia system architecture

A server consists of three modules: a control module, a storage module and a transmission module. The control module is in charge of presenting the catalog of available multimedia contents to the clients, applying the admission control to accept new clients, controling the general activity of the server. The storage module is in charge of transfering multimedia contents between storage system and main memory of the server. The transmission module transfers multimedia

contents from server main memory to the clients. Furthermore, it receives the clients VCR commands controling the multimedia streams.The storage system can be constituted by magnetics disks, video tapes or CD-ROMs libraries. Because of shorter access time, we assume that the storage system is based on magnetics disks.

From the client point of view, the VoD system QoS is characterized by the following requirements:
⋆ (R1) short and upper bounded response time to VCR commands (start, stop, play, jump);
⋆ (R2) fluid visualization of any video content;
⋆ (R3) a minimum interruption of the transmission in case of failures;
⋆ (R4) a various choice of video contents.
Each component of the VoD system contributes to achieve the end-to-end QoS.

1.2 State of the Art

Our main contribution concerns the dimensioning of a SAN based on Fibre Channel in a multimedia system providing VoD services and offering a deterministic Quality of Service. As previously seen, such a system must (i) offer a low latency for VCR commands, (ii) require a small amount of buffers, avoid video starvation as well as buffer overflow, and (iii) support a large number of clients with guaranteed QoS. This goal has been expressed in a lot of papers [6], [7], [8] and [9]. The video striping policy and the scheduling policy are of prime importance to achieve this goal.
With regard to video striping, a classification has been introduced in [3], according to the striping applied to (i) the video content and (ii) on the segment. Each striping can be wide (over all disks of the server), narrow (limited to a subset of disks, for instance the disks connected to a Fibre Channel arbitrated loop) or single (one disk). The combination of a wide video striping and a single segment striping is also called Coarse Grain Striping in [4] and [5]. In the case of a VoD system based on Fibre Channel technology, magnetic disks are connected to arbitrated loops. If we consider an arbitrated loop, the best load balancing of disks connected to this loop leads to split a video content over all the disks of this loop. Moreover, as a disk must first win the loop arbitration before being authorized to transmit the requested data, the technique consisting in striping a segment over several disks loses its interest. That is why a segment is stored on a single disk. We then get the architecture described in section 2.2. The closest work to our corresponds to [3], however it only considers the disk retrieval time and does not account for the time needed to access a shared medium, a Fibre Channel loop in our case. Performance results concerning disks connected to a Fibre Channel arbitrated loop are given in [12] and [13]; these results have been obtained by simulation. We use these results to validate our model in different configurations.

2 Fibre Channel for a Multimedia System

2.1 Fibre Channel Principles

Fibre Channel, FC, is an ANSI standard defining high throughput network technology [1]. Advantages offered by the Fibre Channel technology are:

1. Flexibility. FC technology defines three physical topologies: point-to-point, switched topology also named Fabric and (3) ring topology, also named Arbitrated Loop. These topologies can use optical fibre or copper cable.
2. Performances. FC allows high speed links and several throughputs from 133 Mbps to 1026 Mbps. FC Arbitrated Loop technology allows the interconnection of 127 nodes. FC Fabric allows a maximum of 2^{24} nodes.
3. Load balancing. This technology allows concurrent accesses of servers to the same storage system.
4. Availability. The insertion of a new node can be realized without disconnecting the system. In an Arbitrated Loop topology, each node represents a single point of failure. This drawback can be eliminated by using a hub. Dual loop can be used to tolerate the failure of the medium on one loop.

In this paper, we focus on the arbitrated loop topology. Class 3 is the only possible class on an arbitrated Loop. Class 3 does not require acknowledgements. Frames are used to transfer data.The maximum payload in a data frame is 2112 bytes. To control the data transmission, FC uses control information [1]. For instance, R_RDY, receiver ready, is used for the buffer to buffer flow control, ARB is used to arbitrate the loop access, OPN is used by a Port which owns the loop to initiate a communication with another port on the Loop, and CLS used to finish the communication between two ports on a Loop.

2.2 SAN Based on Fibre Channel in a Multimedia System

A Storage Area Network, SAN, is a high-speed, scalable network of storage devices, servers (connected entities) and interconnecting entities (switch, hub). As in [2], we propose to use Fibre Channel for the SAN. We first present the adopted architecture and then describe how the video contents are stored.

• VoD system architecture
The architecture we propose is based on Fibre Channel Storage Area Network. The storage system is made up by magnetic disks. They are interconnected by means of one or several FC Arbitrated Loops. FC-loops are connected to servers by means of FC-switches. On the other hand, a multimedia system must allow easy extensions when necessary. This situation occurs when for instance the server has to serve a higher number of clients, or the storage capacity must be increased. A modular and flexible architecture is thus necessary. The proposed architecture meets these goals: for instance, we can connect additional arbitrated loops in the storage system without interrupting the system activity.

• **Video content storage**
We assume that the video contents are coded at a constant throughput. Any video content V is stored on all the disks of a loop L. We assume that all the disks in the system have the same block size B. The video content V is split up on all the disks of loop L by m data blocks of size B: one disk contains the first m blocks, another disk contains the m next blocks, where m is a parameter of the storage system introduced to minimize the overhead induced by the disk seek time and rotational latency. Indeed, m blocks of size B are retrieved in a single disk request. We will see in section 3.4 how to determine the best value of m. According to the classification of [3], this configuration corresponds to a narrow striping of the video content and a segment striped on a single disk.

3 Performance Analysis of an Arbitrated Loop

In this section, we establish the feasibility conditions associated with an arbitrated loop of a VoD storage system. We focus on two resources: the magnetic disks and the arbitrated loop. The feasibility conditions are based on the worst case analysis detailed in section 3.1. We show how to apply those results to model the behavior of an arbitrated loop in a SAN system. In this analysis, the feasibility conditions are established between the magnetic disks and the server. To simplify the analysis, we assume that the server connected to the FC fabric encounters a constant delay (no jitter) through the fabric.

3.1 Uniprocessor Real-Time Scheduling

We now focus on uniprocessor real-time scheduling. The results presented here are used in section 3.3, in the context of a storage system based on FC technology. First, we recall some real-time scheduling results for Non-Preemptive Fixed Priority/Highest Priority First (NP-FP/HPF) scheduling. Then we establish the feasibility conditions for sporadic tasks executed with NP-FP/HPF scheduling.

• **Concepts and notations**
We investigate the problem of scheduling a set $\tau = \{\tau_1, ...\tau_n\}$ of n sporadic tasks. We assume that (i) time is discrete and (ii) the times when tasks are requested, are not known a priori. Any sporadic task τ_i is defined by (C_i, T_i, D_i, J_i) with:
⋆ C_i, the maximum execution duration of the task.
⋆ T_i, the minimum interarrival time between two requests of task τ_i, T_i is abusively called the period of task τ_i.
⋆ D_i, the relative deadline of task τ_i. A task τ_i whose activation is requested at time t has $t + D_i$ for absolute deadline, (i.e. it must complete before time $t + D_i$).
⋆ J_i, the maximum release jitter.

⋆ The processor utilization factor, denoted $U = \sum_{i=1}^{n} C_i/T_i$ is the fraction of processor time spent in tasks execution. An idle time t is defined on a processor as a time such that there are no tasks whose activation has been requested

before time t, pending at time t. A busy period is defined as a time interval $[a, b)$ such that there is no idle time in (a, b) and such that both a and b are idle times.
$\star$ We define the following sets: $hp(i) = \{\tau_j, j \neq i, priority(\tau_j) \geq priority(\tau_i)\}$ and $\overline{hp}(i) = \{\tau_j, priority(\tau_j) < priority(\tau_i)\}$. A level-i busy period is a period of activity of the processor where only tasks $\tau_j \in hp(i) \bigcup \{\tau_i\}$ are executed.

We now show how to compute the worst case response times of any sporadic task scheduled NP-FP/HPF. The notion of level-i busy period introduced by [10] for preemptive FP/HPF scheduling is extended. In a non preemptive context, a task τ_i can be delayed by a task τ_j with a lower priority having started its execution before τ_i's release. This priority inversion, called non-preemptive effect, must be accounted for.

• Feasibility and worst case response time computation

Lemma 1. *A necessary condition for the feasibility of any task set is* $U \leq 1$.

Lemma 2. *The worst case response time of any task* τ_i *defined by* (C_i, T_i, D_i, J_i)*, scheduled according to NP-FP/HPF is obtained in a level-i busy period such that (i) all the tasks* $\tau_j \in hp(i) \bigcup \{\tau_i\}$ *are periodic with a release jitter equal to* J_j *and their first occurrence is generated at time* $-J_j$*, and (ii) one task* $\tau_k \in \overline{hp}(i)$ *whose duration is maximum (if any) is released at time* $t = -1$.

Proof. See [11].

Theorem 1. *Let* $\tau = \{\tau_1, ..., \tau_n\}$ *be a sporadic task set scheduled according to NP-FP/HPF. The worst case response time of any task* τ_i *is given by:*

$$r_i = max_{q=0,...Q}\{w_{i,q} + C_i - qT_i + J_i\} \quad (Eq.1)$$

where $w_{i,q} = qC_i + \sum_{\tau_j \in hp(i)} \left(1 + \lfloor \frac{w_{i,q}+J_j}{T_j} \rfloor\right) C_j + max_{k \in \overline{hp}(i)}(C_k - 1)$ $(Eq.2)$

and Q *is the smallest value such that* $w_{i,Q} + C_i \leq (Q+1)T_i - J_i$.

Proof. See [11]. Notice that if $\overline{hp}(i) = \emptyset$, $max_{k \in \overline{hp}(i)}(C_k - 1) = 0$.

3.2 The Scheduling Problem

Before defining the scheduling problem, we first introduce some notations concerning the loop and disk parameters.

• Loop parameters

We consider a loop L. Let N_D be the number of disks in loop L. Let t_{fab} be the delay introduced by the fabric. As already said, this delay is assumed to be constant in a simplifying purpose. N_{dev} is the number of devices connected to a loop. $t_{through}$ is the latency introduced by each device connected to the loop. t_{prop} is the propagation delay on the loop and l_{loop} is the loop latency. The loop latency can be evaluated as follows: $l_{loop} = N_{dev} \cdot t_{through} + t_{prop}$.

Let Th_{loop} be the throughput of the arbitrated loop.
Let t_{ARB}, t_{RDY}, t_{OPN}, and t_{CLS} be the transmission times for respectively an ARB, an R_RDY, an OPN, and a CLS frame.
We now consider a device winning the loop arbitration. When this device asks for the loop arbitration by sending an ARB, it must wait for the receipt of its ARB. Hence a time $t_{ARB} + l_{loop}$.
After having won the loop arbitration, this device starts the communication by sending the OPN, waits for the receipt of a R_RDY and finally sends the frames to be transferred. Hence a time $t_{OPN} + t_{RDY} + l_{loop} + data/Th_{loop}$, where $data$ is the size in bits of the data to be transferred. After the transmission of the last frame, it finishes the communication by sending a CLS and waits for the receipt of a CLS sent by its corresponding device. Hence a time $2t_{CLS} + l_{loop}$.
Let $t_{loopctrl}$ denote the time needed to start and finish a communication on the loop. We then have $t_{loopctrl} = t_{ARB} + t_{OPN} + t_{RDY} + 2t_{CLS} + 3l_{loop}$.

• **Disk parameters**
Let s_{disk} be the seek time of the disk and l_{disk} be the rotational latency.
Let N_C be the maximum number of clients processed by any disk of loop L. We assume that m blocks of size B are retrieved in a single disk request.

• **The scheduling problem**
We want to determine the feasibility conditions associated with the scheduling on disks and on the arbitrated loop. We assume that each disk connected to the loop serves N_C clients and the loop connects N_D disks.
The worst case occurs when all the accepted clients want to retrieve a video content coded at the highest throughput Th_{video}. Let T denote the period of block transmission of a video content coded at the highest throughput. We have $T = B/Th_{video}$. With a period mT, the server generates requests asking each disk D to retrieve m blocks of size B for each of the N_C clients served by D. The disk solicited for a client in a period mT changes every mT.
We assume that the server has a buffer of β blocks of size B per accepted video stream. As soon as m blocks are transmitted to the client, the server asks the disks to retrieve the m following blocks. In order to avoid client starvation, these blocks must be received by the server before the $\beta - m$ remaining blocks in the buffer be transmitted to the client. Hence a deadline equal to $(\beta - m)T$ with the condition $\beta - m \geq 1$.
We assume that the memory available on the disk is sufficient to store the data retrieved from the disk before transmission on the loop. For each client it has to serve, a disk positions the head, reads m blocks and copies them in its memory. It then asks for the loop arbitration to transfer them toward the server. After winning the arbitration, it starts a communication with the server, transfers the requested blocks and then finishes the communication.
On the loop, the server has the highest priority. Each time the server wants to transmit a request, it asks for the loop arbitration. After winning the arbitration, it starts a communication with a disk, transfers the requests to this disk, finishes the communication and then releases the loop. The server proceeds in the same way for each request. The server sends one request per client served by a disk.

The resulting feasibility conditions are expressed in the following, assuming that all the disks serve the same number of clients, this number being the maximum possible for a disk in a given loop configuration.

3.3 Feasibility Conditions

At each period mT, the server sends one request per client served by each disk in the loop L and each disk has to serve N_C clients in this period. Each disk is assumed to use the FIFO scheduling policy. We consider two different feasibility conditions. The first one concerns the condition imposed on the utilization factor of each considered resource (disk and loop). The second one concerns the end-to-end response time for the retrieval of m data blocks for a client of the multimedia system. This time is the time elapsed between the server request time and the reception time by the server of the requested data. This end-to-end response time must meet the deadline as expressed in section 3.2.

• Conditions on the Disk utilization factor
We apply the results given in section 3.1 for sporadic tasks. We first express the fact that for each considered resource, the utilization factor is less than or equal to 1 (see lemma1 in section 3.1). As all the requests coming from a server are sporadic with a period of mT, this condition can be written: the workload on each resource in a period is less than or equal to the period duration.
Each disk must serve N_C clients in a period mT. The service duration of a client is equal to $s_{disk}+l_{disk}+mB/Th_{disk}$. Hence the condition on the disk utilization factor can be written: $N_C(s_{disk} + l_{disk} + mB/Th_{disk}) \leq mT$. We then obtain the maximum number of clients accepted by a disk:

$$N_C \leq \frac{mT}{s_{disk}+l_{disk}+mB/Th_{disk}} \quad (Eq.3).$$

Moreover, the worst case response time of a request is obtained when the N_C requests are received simultaneously by the disk. It is equal to $X_D = N_C(s_{disk} + l_{disk}+mB/Th_{disk})$. The best response time is equal to $s_{disk}+l_{disk}+mB/Th_{disk}$.

• Conditions on the Loop utilization factor
On the loop, we have:
- ⋆ N_CN_D Server tasks of duration $C_{server} = t_{loopctrl} + t_{req}$,
- ⋆ N_CN_D Disk tasks of duration $C_{disk} = t_{loopctrl} + mB/Th_{loop}$.

All these tasks have a period mT and the Server tasks have no release jitter.
The condition on the loop utilization factor can be written: $N_CN_D(C_{disk} + C_{server}) \leq mT$. Hence the maximum number of disks accepted by a loop is given by equation 4:

$$N_D \leq \frac{mT}{N_C(t_{req}+mB/Th_{loop}+2t_{loopctrl})} \quad (Eq.4).$$

• End-to-end response time
We now determine the worst case response time for the response of a disk to the server. The worst case response time between a server request and the receipt by the server of the requested data can be evaluated by means of the holistic approach [14]. This response time consists of three parts X_R, X_D and X_L where:

⋆ X_R is the latest reception time of a server request by a disk,
⋆ X_D is the disk worst case retrieval time,
⋆ X_L is the additional worst case time needed to transfer the requested data to the server (including loop and fabric transfer).

X_R can be expressed as follows: $X_R = t_{fab} + N_C N_D t_{loopctrl} + t_{req} N_C N_D + mB/Th_{loop} + t_{loopctrl}$, where $t_{loopctrl} + mB/Th_{loop}$ corresponds to a non-preemptive blocking factor due to a disk having just started its transmission on the loop when the server decides to transmit the disk requests.

X_D can be expressed by considering the last client served by a disk in a period of duration mT. We have: $X_D = N_C(s_{disk} + l_{disk} + mB/Th_{disk})$ for the FIFO policy.

In the worst case, time X_L is obtained considering a disk that gains the loop arbitration after the other disks. The server and the disks have tasks of period mT. We also consider that the server generates its requests with no release jitter, and the disks receive the requests with a jitter J_{disk}. We now determine this maximum jitter J_{disk} for a disk accessing the loop:

⋆ In the worst case, the demand to transmit on the loop the server request for a client is processed after a delay$X_R = t_{fab} + N_C N_D C_{server} + C_{disk}$. The first term is due to the fabric, the second one means that this demand is the last one among the $N_C N_D$ demands to be served. The third term corresponds to the non-preemptive effect: when the server asks for the loop transmission, a disk has just started to transmit its m blocks. According to the FIFO scheduling, this request is served after a maximum delay of $X_D = N_C(s_{disk}+l_{disk}+mB/Th_{disk})$. Hence the read blocks are ready to be transmitted on the loop after a maximum delay of $X_R + X_D$.

⋆ In the best case, the demand to transmit on the loop the server request for a client is processed after a delay $t_{fab} + C_{server}$. The disk has read the requested blocks after a delay $s_{disk} + l_{disk} + mB/Th_{disk}$. Hence the read blocks are ready to be transmitted on the loop after a delay of $t_{fab} + C_{server} + s_{disk} + l_{disk} + mB/Th_{disk}$.

⋆ The disk jitter J_{disk} is obtained by the difference between the worst case and the best case: $J_{disk} = (N_C N_D - 1)C_{server} + C_{disk} + (N_C - 1)(s_{disk} + l_{disk} + mB/Th_{disk})$.
We can apply theorem1 to the considered disk, to compute $w_{disk,q}$ the latest starting time of the q^{th} iteration of a Disk task:
$w_{disk,q} = q \cdot C_{disk} + N_C N_D(1 + \lfloor w_{disk,q}/(mT) \rfloor)C_{server} + (N_C - 1)(1+q)C_{disk} + (N_D - 1)N_C(1 + \lfloor (w_{disk,q} + J_{disk})/(mT) \rfloor)C_{disk}$.

⋆ In the formula giving $w_{disk,q}$ the first term stands for the workload induced by the considered client served by this disk in the q previous iterations and the second term represents the workload induced by the server. The third term stands for the workload induced by the $(N_C - 1)$ other clients of the considered disk and the fourth term accounts for the workload induced by the N_C clients of the $N_D - 1$ other disks.

$\star$ The stop condition is given by Q the smallest integer value such that $w_{disk,Q} + C_{disk} \leq (Q+1)mT - J_{disk}$.

We then get: $X_L = t_{fab} + max_{q=0..Q}(w_{disk,q} - qmT) + C_{disk}$. According to the holistic approach, the end-to-end response time can be upper bounded by $X_R + X_D + X_L$. We now express the constraint related to the end-to-end deadline $(\beta - m)T$, leading to $X_R + X_D + X_L \leq (\beta - m)T$. We finally get:

$$N_C N_D(t_{loopctrl} + t_{req}) + 2(mB/Th_{loop} + t_{loopctrl} + max_{q=0..Q}(w_{disk,q} - qmT) + N_C(s_{disk} + l_{disk} + mB/Th_{disk}) + 2t_{fab} \leq (\beta - m)T \quad (Eq.5).$$

3.4 Model Validation and Performance Results

In this section, we compare the performance results obtained in our analysis with simulation results already published in [12] and [13]. These comparisons are made for different configurations. After this validation, we study the influence of different parameters on the performance of the VoD system. The maximum coding throughput considered for the video contents stored in the multimedia system is equal to 3 Mbps. In all the experiments, the size B of the block is equal 64 kBytes, leading to $T = 174.7ms$. We consider different values of m. In all graphs, we represent the total useful throughput as a function of the number of disks in the loop. The number of clients accepted by the VoD system is equal to the total useful throughput divided by the maximum coding throughput of video contents. In our experiments, we use three types of disks, whose parameters are given in the following table. Disks D_1 and D_2 are Seagate disks used in [12]. Disk D_3 is an IBM Ultrastar XP disk used in [13].

	D_1	D_2	D_3
seek time (ms)	10.5	8.34	8.5
rotational latency (ms)	5.5	4.15	4.17
sustained throughput (Mbps)	33.6	58.8	52.04

• Model validation

We represent the total useful throughput obtained on the loop considering different disk numbers, different disk parameters and different sizes of block. We compare our results with the results of [12] in figure 2a and the results of [13] on figure 2b. On figure 2a $m = 2$, leading to the retrieval of 128 kBytes for every read access. On figure 2b, $m = 1$. The results are very close and show that our model is valid for different configurations.

• Influence of the deadline

In this experiment illustrated by figure 3, $m = 2$, and the deadline is equal to $2T$, $4T$, $6T$ and $8T$. As long as the number of clients meets Eq3 and the number of disks meets Eq4, an increase of the deadline makes easier Eq5 and therefore improves the maximum number of accepted flows. For a small number of disks (less than 13 in figure 3), the deadline influence is not significative. Indeed, Eq5

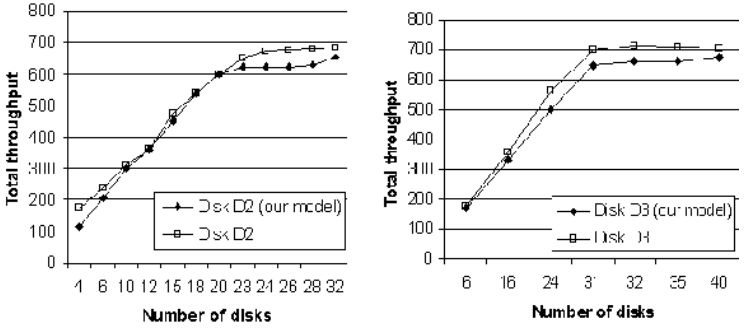

Fig. 2. Model validation a) for disk D_2 and b) for disk D_3

that is the only equation accounting for the deadline, is not the limiting one. For a higher number of disks, a deadline increase improves the performances. However, a deadline higher than $6T$ does not improve significatively the number of accepted clients.

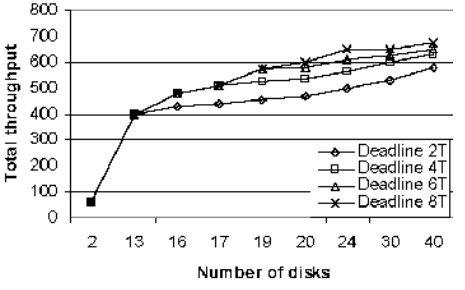

Fig. 3. Influence of the deadline

Moreover, there is a limitation imposed on the server buffer size which determines the latency for a play command. Indeed, the server buffer reserved for a client must be filled before the video content visualization starts. Hence, the optimal size is determined as a trade-off between the maximum latency acceptable by a client and the maximum number of flows accepted by a VoD system.

• **Influence of disk performance**
In this experiment illustrated by figure 4a, $m = 2$, the deadline is equal to $6T$. This experiment shows the influence of disk performance on the number of accepted clients. In Eq3 and Eq5, the best performance of the VoD system is obtained for disks providing the smallest value of $s_{disk} + l_{disk} + mB/Th_{disk}$.

This is achieved for disk D_3. A high performance disk is more interesting when the number of disks is less than 30. Over this threshold, the loop becomes the limiting factor.

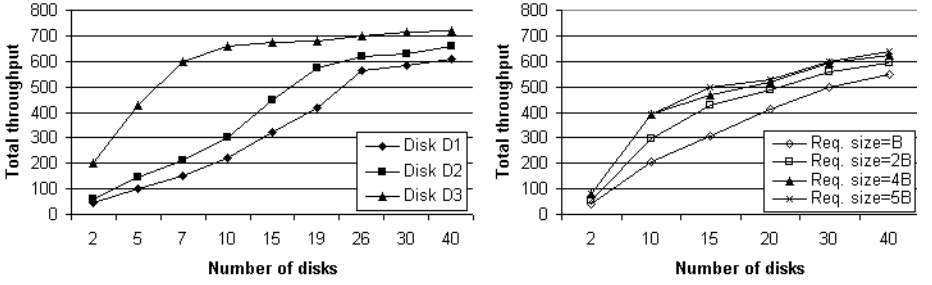

Fig. 4. Influence of a) the disk performance and b) the number of read blocks

• **Influence of the number of read blocks**
In this experiment illustrated by figure 4b, we consider different values of m (m = 1, 2, 4 and 5), the deadline is equal to mT. A value of m higher than 4 does not significatively improve the number of accepted flows and the useful throughput. With $m = 4$, we have a deadline equal to 698.8 ms which is acceptable for the response time of the play/start command. Nevertheless, a high value of m is not necessarily suitable as it influences the server buffer size and the disk buffer size.

Conclusion

In this paper, we have proposed to use Fibre Channel technology in multimedia systems offering Video on Demand services and ensuring a deterministic QoS. A SAN architecture offering a good scalability has been defined. We have shown how to dimension the arbitrated loops of the SAN. This dimensioning is established from a uniprocessor real-time scheduling analysis applied to two crucial resources: magnetic disks and FC arbitrated loop. Our analysis has been validated by comparing our results with previously published simulation results. We have then computed the maximum number of clients acceptable by a loop, depending on the disk (number and performance). The optimal number of disks connected to a loop has been determined. Our analysis can be used as a dimensioning tool for a VoD system. More precisely, the results can be used to implement an admission control for new VoD clients.

References

1. A. F. Benner, "Fibre Channel for SANs", McGraw-Hill, 2001.
2. S. Wilson, "Managing a Fibre Channel Storage Area Network", http://www.sansolutions.com/SNMWG/Downloads/SAN_White_Paper-V.05.pdf
3. J. Gafsi, "Design and performance of large scale video servers", Ph. D Thesis, ENST Paris, France, Nov. 1999.
4. S.A. Barnett, G.J. Anido, P. Beadle, "Predictive call admission control for a disk array based video server", Multimedia Computing and Networking, San Jose, California, Feb. 1997.
5. B. Ozden et al., "Disk striping in video server environments", IEEE Conf. on Multimedia Systems, Hiroshima, Japan, June 1996.
6. D. R. Kenschammaana-Hosekote, J. Srivastava, "I/O scheduling for digital continuous multimedia", Multimedia Systems 5, pp. 213-237, 1997.
7. S. Sengodan, V. O.K. Li, "A quasi-static retrieval scheme for interactive VOD servers", Computer Communications, 20, pp. 1031-1041, 1997.
8. H. M. Vin, A. Goyal, P. Goyal, "Algorithms for designing multimedia servers", Computer Communications, 18(3), pp. 192-203, March 1995.
9. R. Wijayaratne, A. L. N. Reddy, "Integrated QoS management for disk I/O", IEEE Int. Conference on Multimedia Computing and Systems, pp. 487-492, Florence, Italy, June 1999.
10. J.P. Lehoczky, "Fixed priority scheduling of periodic task sets with arbitrary deadlines", Proc. of 11th IEEE Real-Time Systems Symposium, Lake Buena Vista, FL, USA, pp. 201-209, Dec. 1990.
11. L. George, N. Rivierre, M. Spuri, "Preemptive and non-preemptive real-time uniprocessor scheduling", INRIA Rocquencourt, RR 2966, France, Sept. 1996.
12. S. Chen, M. Thapar, "Fibre channel storage interface for video-on-demand servers", Proc. of Multimedia Computing and Networking, San Jose, CA, Jan. 1996.
13. D.H.C. Du, J. Hsieh, T. Chang, Y. Wang and S. Shim, "Performance study of serial storage architecture (SSA) and fibre channel arbitrated loop (FC-AL)", Computer Science Dept., Univ. of Minnesota, TR96-074, 1996.
14. K.Tindell, J. Clark "Holistic schedulability analysis for distributed hard real-time systems", Microprocessors and Microprogramming 40, 1994.