Modeling of multimedia architectures: the case of visioconferencing with guaranteed quality of service

PHILIPPE OWEZARSKI – MARC BOYER

1. Introduction

Computing multimedia data changed radically the way applications must be developed: computers do not have only to perform as many computing as possible as fast as possible (best effort), but also they must present each information unit at a time which respects its temporal validity.

This change forces us to rethink the way applications are designed, and to develop models able to consider temporal constraints of distributed multimedia applications.

In this chapter, a visioconferencing system with guaranteed quality of service is presented for this purpose. It is exhibited how the use of a formal model (Time Stream Petri Nets, presented in section 6) helps, from the modeling of multimedia stream constraints, to propose a synchronization constraints architecture, which leads the global conception of the application.

Section 2 presents characteristics and requirements of multimedia information and asynchronous systems which are the hardware and software supports on which multimedia applications are now designed. Section 3 shows how Time Stream Petri Nets allows the modeling of multimedia flows constraints; from this modeling of multimedia constraints, a synchronization architecture of the application is derived (section 4).

2. Problematic of multimedia synchronization

2.1. Multimedia information: characteristics and requirements

2.1.1. Notion of flow, continuous flows, discrete flows

The first characteristic which makes multimedia data different from classical computer data (text, binary data, …) is their computing unit: multimedia data consist of ***flows***, while texts and binary data are handled as files. Indeed, audio and video data are sequences of images or audio samples which succeed each other with a constant or not rate. Multimedia data is not incompatible with the notion of computer file, and files are still used to save movies or audio documents; but the computing are made information unit by information unit (image by image, for instance); these information units, back to back, form a flow. Note however that texts, graphics and binary data can take advantage of this kind of computing.

However, flows are also characterized by the temporal relationships which exist between the different information units. For example, it does not exist any temporal relationship between the characters which form a textual flow. Similarly for fix images which can be considered as bit flows, or for graphics, it does not exist any temporal relationships between the different units of the flow. These are typically ***discrete flows***.

At the opposite, for video or audio, images or audio samples must be produced, computed and presented at a regular rate. These are ***continuous flows*** (or ***streams***). If the time interval between two consecutive flow units is constant, the flow is said *isochronous*; nevertheless, a given variability on these time intervals can be tolerated: this variability is called *authorized jitter*.

2.2.2. The notion of quality of service

In the presentation of the flow notion, just over, already appeared the notion of temporal constraints. However, many other quality parameters exist. In fact, multimedia flows are characterized by their quality of service (or QoS). One of the QoS parameters families deals with the presentation of multimedia flows.

For example, binary or textual data does not stand any loss (they require a full reliability); at the opposite, their requirements in terms of storage capacity or bandwidth for being sent on a network are low. In addition, this kind of data has no temporal constraint.

Similarly, graphics hardly stand losses or errors. They require only very limited storage or communication resources, and are not really sensitive to jitter phenomenon.

On the other side, fix images can support some error or loss rate. However, the required storage capacity and network bandwidth are higher than for texts or graphics. Besides, in order to reduce the amount of data of a video image, compression algorithms have been designed as FIF or JPEG[1] [WALLACE 91].

Audio flows can hardly stand errors or losses; such issues decrease dramatically at the quality perceived for the audio flow. The required throughput and the storage capacity are variable according to the used coding: only 64 kbps are required for digital telephone quality, while 1.2 Mbps (without compression) are required for HiFI laser quality. In addition, the audio flow is very sensitive to temporal disturbances.

Finally, live video is the less constrained media in terms of errors or losses: the loss of one image is not perceived by the end user. At the opposite, the amount of data such a flow represents and the throughput required are very huge. Besides, many efforts have been done in the area of video compression with algorithms as H261 [LIOU 91] [TURLETTI 93], MPEG[2] [Le Gall 91], etc.; However, video streams, even compressed, remain very demanding in memory space and bandwidth. In addition, temporal constraints on flow images are strong, even if a given jitter level can be acceptable.

In what precedes, we just did a small summary of the quality of service constraints for different kinds of media. However, this summary is far from being exhaustive and does only consider QoS parameters related to losses, errors, required memory space and bandwidth. Anyway, enumerating all QoS parameters is impossible as each application has its own requirements, and will then have to fulfill its own QoS parameters on its own data. We just presented the most common QoS parameters in current multimedia applications as visioconferencing.

2.1.3. Multimedia synchronization

Previous part just exhibited the multimedia QoS notion and illustrated it by presenting the main QoS parameters associated to multimedia flows. However, the temporal aspect which has been signaled in part 2.1.1 is essential, and continuous flows are basically characterized by their temporal constraints. For example, in a multimedia presentation which includes several kinds of data, temporal and spatial relationships inside the flows exist. It can also exist temporal relationships between these flows. These relationships define multimedia synchronization constraints. The following presents its different aspects.

2.1.3.1. Spatial synchronization

[1] JPEG: Joint Photographic Expert Group.
[2] MPEG: Motion Pictures Expert Group.

Spatial synchronization expresses the visual scheduling constraints of the different multimedia objects on the presentation support (screen, image wall, etc.). It allows the definition of the size of the different areas, overlays, juxtapositions, etc. as in the MHEG[3] standard [ISO 90] [ISO 93].

2.1.3.2. Temporal synchronization

Solving the problems related to temporal synchronization is the most important and difficult to solve in the design of multimedia distributed applications and systems [BLAKOWSKI 96]. It aims at expressing and guaranteeing temporal constraints and relationships which exist between the constituting objects of a multimedia document. Two kinds of synchronization appear: intra and inter-stream synchronizations.

Intra-stream synchronization aims at enforcing the presentation constraints on each information unit of the flow; this consists in controlling the jitter in order to make the presentation duration not diverge from its ideal value by a value greater than the maximum authorized jitter. For example, for a live video, we must guarantee that the presentation time of each video object (image) respects the time that existed between two consecutives image captures when the movie was created.

Inter-stream synchronization aims at controlling the temporal drift that can exist between two streams. Drift is due to the cumulative effects of jitters; indeed, jitters supported by the objects can cumulate, and the drift can become very high. It is then required to control this drift in order to make it remain under a tolerance threshold. This is typically the case in the synchronization between one audio and one video streams, for which it is required to enforce that sounds correspond to the lip movements; this issue is known as the ***lip synchronization*** problem.

Similarly, it exists a difference between discrete and continuous synchronization. Discrete synchronization appears most of the time when only discrete flows are considered, it consists in synchronizing objects when it is necessary. For example, in the case of a movie with sub-titles, it is required to synchronize subtitles with the movie only when there are dialogues.

On the other side, continuous synchronization consists in periodically introducing synchronization points in the presentation of flow(s). For example, in the case of lip synchronization, it is necessary to periodically introduce synchronization points between the audio and video streams to avoid the drift to increase too much.

2.1.3.3. Hypermedia synchronization

[3] MHEG: Multimedia and Hypermedia information coding Expert Group.

Hypermedia synchronization integrates spatial and temporal synchronization notions. However it also adds the notion of logical synchronization which consists in synchronizing the presentations of the application according to the activation of links of an hypermedia document (as in MHEG [ISO 90] [ISO 93]). Hypermedia synchronization will however not be considered in this chapter.

2.2. Asynchronous distributed systems

2.2.1. Why asynchronous distributed systems?

The study and methodology described in this chapter rely on the use of asynchronous distributed systems, and this for three reasons:

- First, almost all distributed systems are asynchronous. LANs, Internet and operating systems (Unix, DOS, …) are asynchronous. Moreover, current trends (imposed by manufacturers) develop asynchronism to improve performances, and asynchronism seems to be the future of computers. As a consequence, this study has been made in the widest possible domain, to propose synchronization mechanisms that can be generalized for a maximum number of applications on a maximum number of platforms;
- Then, the solution of the multimedia synchronization problem in a synchronous environment is a particular case of the one in asynchronous environment, where asynchronism is null, and we then provide a solution for the most general case;
- Finally, with the real-time scheduling classes in Unix (as in Solaris 2 for instance), it becomes possible to implement real time applications (this was not the case before) [COULSON 94] [JEFFAY 92] [JEFFAY 94B] [VOGEL 95]. However, the gap between the theory of real-time system and the reality of experimental real-time available systems is large [KATCHER 94] Indeed, using real-time operating systems means to solve some of the same problems as with asynchronous supports because :
    - inputs / outputs are asynchronous ;
    - priority inversions can appear when synchronizing processes ;
    - the kernel has to be fully preemptive ;
    - system tasks can disturb the temporal scheduling of applications ;
    - etc. …

In fact, for a real-time behavior with a real-time operating system, it is required to limit the real-time processes to operations that do not perform any inputs / outputs operations [BAKER 94]. However, this is incompatible with distributed multimedia applications that have to access multimedia and network boards. On the other hand, using classical systems (UNIX or POSIX) is of high interest, because they are widely available, and they imply only a few more problems than using "pure" real-time systems [ADELBERG 94]. In addition, [KANG 94] showed that high speed wide area synchronous systems (including communication supports and operating systems) are impossible to realize, thus justifying the use of asynchronous systems.

2.2.2 Characterization of asynchronous systems

But, using asynchronous bases to support isochronous multimedia data introduces several problems. The essential one follows from the temporal variability of computing, as a real asynchronous operation has no upper bound. This variability (asynchronism) appears at three levels within any distributed system.

- Communications supports and protocols are asynchronous. For example, the media access modes of most networks are non deterministic (depend on network load for instance), and no upper bound on the transit delay is ensured. This phenomenon is more important when using a WAN like Internet.
- In a classical operating system (as Unix), variability is due:
  - to the time shared scheduling mechanisms (to privilege interactive processes and average throughput instead of temporal constraints). Moreover, the heavyweight processes notion, as in Unix, introduces a scheduling overhead that prevents parallelism and high throughput required by multimedia streams;
  - to the non preemptive or locally preemptive nature of the kernels, that induces non deterministic latency times for process switching or interruption handling;
  - to memory swapping and other system tasks that run at non deterministic instants and with priorities greater than the ones of users tasks;
- The audio and video boards are not synchronous. In fact, new processors integrate multi-level cache memories for instructions and data that generate non deterministic memory access and context switching time. Likewise, virtual memory introduces a variability in memory access time.

2.2.3. Problems due to asynchronous systems

The problems which appear in asynchronous systems then are:

- Jitters problems due to the temporal variability of operations run in the kernel. This later cannot guarantee a constant computing time and/or predictable for the different operations, and then cannot guarantee the presentation times of multimedia data;
- Drift problems which are due to the cumulative effect of jitter, and can make a large drift appear, especially after the computing of a long sequence presentation;

- Finally, loss and duplication problems; Indeed, the scheduling of task being unpredictable, it is possible that in the case of a buffer management, for instance, production tasks get easier access to the processor than consumption tasks. This would lead to a buffer overflow and losses. Similarly, losses can appear in communication networks. Duplicated objects can also arise, in particular in networks, when the routing algorithm duplicates packets.

All these issues have to be solved, or at least controlled, in order to make possible writing multimedia distributed applications.

3. Modeling of multimedia synchronization constraints

3.1. Modeling requirements

Previous section 2 points out the variability of possible QoS constraints. In addition, temporal constraints are difficult to express, and one of the first issue to address deals with easily and completely representing synchronization constraints which can exist in a multimedia document.

This pojnt did not appear as essential at the beginning of our work, and we thought that it would be possible to guarantee intra and inter-streams synchronization in a visioconferencing system without modeling the constraints to be enforced. For this purpose, an intuitive solution for synchronization deals with using timestamps [DIAZ 94A] which indicate the presentation date of each object. Thus, each object is timestamped with a relative date (0 being the start of the application), and the presentation process just has to present the considered object at the time indicated by its timestamp. This technique was used for developing a first visioconference prototype called TSVS (Timestamp Synchronized Visionconference system)[4], whose pros and cons are evaluated in what follows (this technique was then re-used in [ROTHERMEL 95]).

The timestamp based synchronization technique is easy to implement, and it reduces both intra and inter-streams synchronization issues. Intra-stream synchronization is obviously enforced as a sequence captured in N seconds is replayed in N seconds with regular presentation rate of objects. On the other hand, inter-streams synchronization is also enforced as each stream is synchronized

[4] TSVS is a visioconference system synchronized thanks to the timestamp technique. The synchronization quality is very good and the tool works very well when problems related to asynchronism are limited (i.e. machines and networks are not loaded). This tool is available upon request to the authors.

according to the common real timeline what synchronizes both streams to each other [DIAZ 94A].

However, timestamps do not take into account the asynchronism notion. They also do not consider the notion of temporal intervals, and then cannot support any jitter on an object. Therefore, if an object arrives after its presentation date (even if this delay is very short), it will not be presented; this creates a discontinuity in the stream whereas the delay would maybe correspond to an acceptable jitter; therefore, some data were discarded whereas it could have been presented after a short delay[5]. The QoS degradation is then much stronger than what it could have been, what is not acceptable if the objective is to enforce the best possible presentation quality. In addition, because of the asynchronism, it is impossible to guarantee any fix date at operating system level. Last, the timestamp based synchronization technique suffer from an a posteriori knowledge of synchronization constraints to be enforced; the end synchronization entity only knows the presentation date of any object when it effectively receives this object; this end entity then cannot anticipate the computing to achieve.

These statements confirmed that it is mandatory to model very accurately multimedia synchronization constraints that need to be respected by our applications. For this purpose, we have been looking at previous work in the state of the art to find out a model which could match our requirements.

To design guaranteed synchronized multimedia applications in asynchronous environment, it is mandatory to consider all problems due to the temporal variability of the computing times (jitter, drift) and to link them to the properties inherent to each multimedia object (for instance to link the jitter of an operating system to the acceptable jitter of each multimedia object). To define these synchronization properties on the multimedia objects themselves, a model allowing the author of a multimedia application to model the application synchronization constraints is required. This formal approach is also very interesting as it allows computer scientists to also simulate and validate the modeled scenarios. Several studies have already been realized in this domain, and some models have been proposed [DIAZ 93A]. In particular, some of these models use formal approaches based on time Petri nets whose graphical characteristic is close from the paradigm of the digital VCR [SÉNAC 94]. Chapters 5 and 6 present the state of the art of temporal

---

5 In TSVS, jitter problems are solved using an ad hoc method. Indeed, for avoiding positive jitter issues (which cannot be solved), received data are buffered for a duration greater than the maximum jitter generally observed in the systems and networks. However, this buffering time is impacting the interactivity level as it leads to some significant delays between the sender and the receiver. That is why TSVS works well only when issues related to asynchronism are limited.

extensions of Petri nets, especially confronting them to the multimedia synchronization problematic in terms of modeling and expressivity capabilities[6]. From this state of the art, it appeared that multimedia synchronization models and temporal extensions of Petri nets which have been proposed up to now do not provide the required expressivity and modeling capabilities for specifying synchronization scenarios of multimedia applications.

The limitations observed on the existing models led us to propose a new model (based on Petri nets[7]), called TSPN (Time Streams Petri Nets, and presented in chapter 6), providing the expressivity capability of the TPN model and the modeling capability of the ATPN model. The TSPN [DIAZ 93A] [DIAZ 93B] [SÉNAC 94] model extends the ATPN model by adding inter-streams firing rules on transitions.

By definition, TSPNs use temporal intervals on the arcs leaving places. This allows the authors to take into account both the temporal non determinism of distributed asynchronous systems and the presentation time variability of multimedia objects. The temporal intervals are triplets (x^S, n^S, y^S) called validity time intervals, where x^S, n^S and y^S are respectively the minimum, nominal and maximum presentation values. Nominal values are useful for computing temporal drift on arcs (compared to the nominal duration).

The inter-streams temporal drifts can be controlled in a very precise way using 9 different inter-streams transition semantics. From an execution point of view, this synchronization semantics are defined as synchronization instants taking into account the real duration of processes; from a modeling point of view, this firing rules define firing intervals considering all possible synchronization instants, obtained by a complete combination of dynamic temporal validity intervals of considered arcs [DIAZ 93A] [SÉNAC 94]. For example, by using these transition rules, it is possible to specify synchronization mechanisms driven by the earliest stream ("or" synchronization rules), the latest stream ("and" synchronization rules) or by a given stream ("master" synchronization rules). This synchronization semantics defines the synchronization instants from an arc statically or dynamically chosen.

3.2. Modeling example for a visioconference application

---

[6] The modeling capacity of model is its ability to easily represent a scenario. Its expressivity capability is its ability to specify completely the scenario.

[7] Because of the graphical aspect of Petri nets which permit a better visualization synchronization characteristics of a multimedia document.

The TSPN model is perfectly suited for modeling synchronization constraints of multimedia streams, in an asynchronous environment, for applications we are considering. Thanks to its high modeling and expressivity capabilities, this model makes possible to easily and completely model complex synchronization scenarios. In addition, from the description in TSPN of a synchronization scenario, it is possible to check its temporal validity. This verification is possible thanks to techniques designed for this purpose, but whose description is out of the scope of this chapter[8].

Concerning the multimedia synchronization aspect in this chapter, the key point deals with studying how it is possible from a TSPN to describe the behavior of a synchronization layer, and in particular, to get the temporal scheduling of multimedia presentation processes for a presentation with all intra and inter-streams synchronization respected.

The TSPN model will be used in the following to describe and implement the synchronization constraints of a visioconference application. Besides, the visionconference case perfectly exhibits the TSPN model capabilities.

In a visioconferencing application, a video and an audio streams have to be synchronized. Some of the dynamic QoS parameters can be described by a TSPN. Figure 1 models a visioconference application for which dynamic QoS parameters are:

- Throughput: 10 images per second;
- Acceptable jitter on an audio or video object: 10 ms [JEFFAY 94A];
- Audio part is the most important media (because the sound is the media that contains most information);
- Synchronization quality: the inter-stream drift must not overrun 100 ms [JEFFAY 94A], 100 ms being a limit under which the temporal gap between audio and video cannot be heard.

These QoS parameters determine the parameters of the TSPN presented on Figure 1, where:

- the 10 images per second rate defines the nominal presentation time of a video object, i.e. 100 ms (if we consider identical granularities for both audio and video, it is also the nominal presentation time of an audio packet);
- the maximum acceptable intra-stream jitter determines the temporal validity intervals that are [90, 100, 110];
- the inter-stream synchronization is of "and-master" type for the sound. In fact, the sound being more important than the video, its temporal constraints have to

[8] [COURTIAT 96] also proposes a methodology for verifying the temporal validity of a multimedia / hypermedia document.

be always respected, even if the ones on video are violated. However, the objective is to synchronize two continuous streams the one from the other, and as much as possible avoid discontinuities on the video stream (which could be caused by the acceleration mechanisms of the and type firing rule). For this purpose, the “and-master” firing rule has been selected, guaranteeing that the constraints on the audio stream will be guaranteed, and trying as much as possible to respect the temporal constraints on the video stream too, when possible;

- the inter-stream drift must not overrun 100 ms; the inter-stream synchronization period corresponds to the presentation of 5 images. Indeed, the maximum drift on 5 audio or video objects is 50 ms. The inter-streams drift between the two streams is then at most 100 ms;

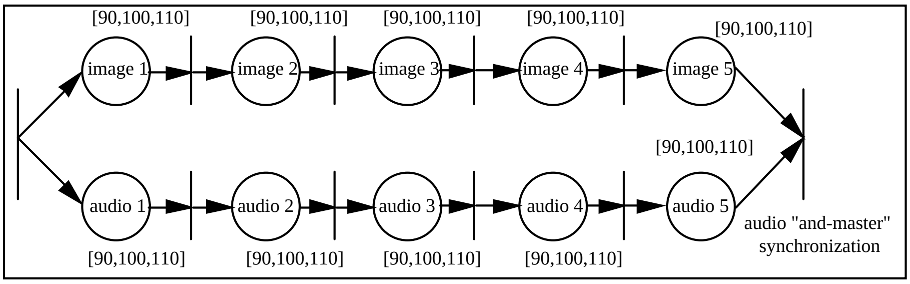

Figure 1. *TSPN example for visioconferencing at a 10 images/s rate*

4. Modeling of a synchronization architecture

4.1. Introduction

This section aims at studying and developing an approach and a set of mechanisms for guaranteeing to users some multimedia QoS parameters as audio and video quality, video rate, end to end delay or temporal synchronization constraints for distributed multimedia applications. More specifically, this part focuses on how to guarantee synchronization constraints of a visioconference application in an asynchronous environment (PNSVS: Petri Net Synchronized visioconference system). Indeed, multimedia synchronization is the key constraints to be enforced for multimedia distributed systems [BLAKOWSKI 95]. The problematic related to multimedia synchronization, as detailed in section 2.1.3 of this chapter, aims at enforcing both intra and inter-streams synchronization.

In the following of this part will address the following points: first, it will be shown that the behavior of the synchronization application can be very different from the one expressed by the users, in order to take into account the specific characteristics of the computer hardware devices and of the operating system. This part (4.2.2) will show that two levels of synchronization exist in a visioconference application as PNSVS. Then, by analyzing results get with PNSVS, section 4.3 will show that it is interesting to use for PNSVS a partial order transport service, which, compared to a standard transport, to significantly improve performances and the quality of presentation. The new architecture of PNSVS will then be presented and evaluated in section 4.3.5.

4.2. Modeling of a visioconference application

4.2.1. Multimedia boards latency and inter-stream synchronization: the drifted rendezvous

The presentation TSPN depicted on Figure 1 shows how units of the audio and video streams have to be synchronized. In particular, for a normal inter-stream synchronization, any audio object i must be synchronized with image i. Nevertheless, the multimedia boards do not have the same latency time. If the video board has a 50 ms latency time and the audio board a 250 ms latency time, then, by respecting the presentation TSPN of Figure 1, the final presentation will not be synchronized: the audio part will be 200 ms late compared to the video part, i.e. audio object i would be synchronized with image i+2, even if the synchronization mechanisms have synchronized the audio object i with the image i.

To solve the problem of those different latency times on different multimedia boards, drifts have to be introduced in the inter-streams synchronizations (we also call this drifts “drifted rendez-vous” [OWEZARSKI 96a]). Indeed, the difference between the audio and the video board latency times being 200 ms, it is sufficient to synchronize audio object i with image i-2 (see Figure 2 which represents the applicative TSPN modeling the applicative behavior of the application): after having been computed by the presentation boards, audio object i will be synchronized with image i.

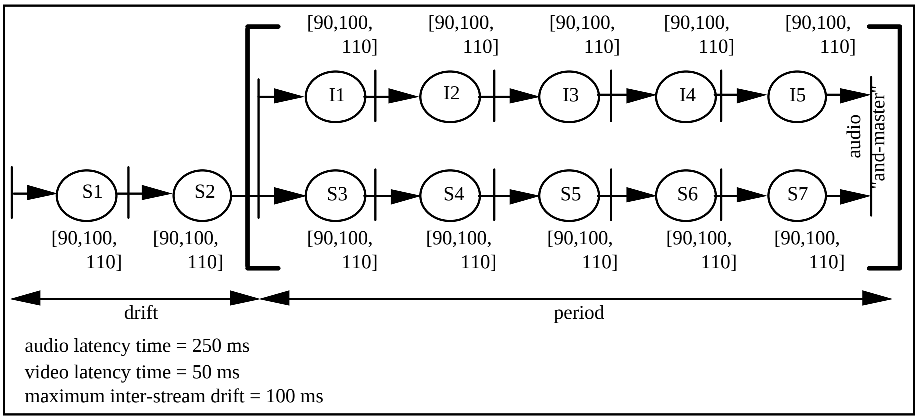

Figure 2. *Applicative TSPN taking into account the hardware latency times*

However, the chosen example is simple as the difference between the two latency times is a multiple of the presentation time of an object. Of course, this is not always the case as for instance if the audio and video latency times equal respectively 230 and 50 ms. The difference between the latency times is 180 ms: the drift that has to be modeled in the rendezvous corresponds to two drift objects, that represents a 200 ms drift. Nevertheless, after this 200 ms drift, there is a 20 ms remaining drift between the audio and video streams. To enforce the maximum inter-stream drift to remain under 100 ms, the TSPN inter-stream synchronization period has to be changed. If the inter-stream synchronization is enforced after every 5 images, because of the 20 ms remaining drift which was not canceled, the inter-stream drift of the audio/video streams would be in the interval [-80 ms, 120 ms], what can be greater than the allowed 100 ms in absolute value. It follows that an inter-stream synchronization must be done after every 4 images, in order to make the inter-stream drift remain in the interval [-60 ms, 100 ms].

The presentation TSPN modeling the multimedia synchronization scenario and the applicative TSPN modeling the applicative processes behavior have different shapes, because of the drifted rendezvous and the modified inter-stream synchronization period.

4.2.2. The two levels modeling of PNSVS

It now clearly appears that two models are needed to describe both presentation and application levels. The first is the interface level whose behavior is modeled using the presentation TSPN. This TSPN models the application behavior as seen from the users, i.e. how audio and video objects are synchronized and what are their temporal constraints. This presentation TSPN is the same for the two users: the sender and receiver have to compute the same objects having the same temporal constraints (cf. Figure 1).

The second level corresponds to the applicative synchronization. The applicative synchronization cannot be modeled by the presentation TSPN because of the characteristics of the operating system and of the multimedia boards. These characteristics force the synchronization mechanisms to consider multimedia objects in a different way than at the interface (presentation) level. To model the applicative synchronization mechanisms, the applicative TSPN model has been defined. Moreover, the sender and receiver entities appear to be different at the application level, and two models are required. Figure 2 represents the applicative TSPN for the receiving entity of PNSVS[9]. [OWEZARSKI 96b] gives the full definition of both TSPN models and shows how the applicative TSPNs are automatically derived from the presentation TSPN, which is itself derived from the QoS requested by users.

4.2.3. Software architecture for synchronization

In asynchronous distributed systems, all components are asynchronous: communication supports, operating systems, multimedia boards. Therefore, synchronization operations have to run at the highest level of the visioconference system architecture, i.e. in the application layer of the receiver. Indeed, enforcing strong (final) synchronization operations within the communication layers would often be useless because the data synchronized in the low layers can be desynchronized when going through upper layers: operating system and multimedia boards can introduce a non deterministic jitter in the computing of each multimedia object, thus annihilating the effect of low layers temporal synchronization processes. In addition, the application layer (user part of the operating system) is the only layer where developers can change the process scheduling.

However, the application written in the user space of the operating system has to be weakly synchronous, and has therefore to respect maximum presentation times. The operating system being asynchronous, no bound on computing times is ensured in the system and time-shared scheduling classes. To be able to assure presentation duration we will not be greater than their maximum bound, it is required to run processes which priorities greater than the ones of the system tasks, and to use a

[9] Note however that the TSPN of Figure 2 can still be modified. We already shoed that inter-streams synchronization constraints can lead to reducing the inter-streams synchronization period. On the other hand, because of the behavior of audio devices we can have to reduce the size of audio packets in order to limit end to end delays [OWEZARSKI 96b] [OWEZARSKI 98a]. Besides, on the global modeling of the visioconference system on Figure 7, it appears that audio packets have been divided by two for this purpose. The production time of audio data is then divided by two, as well as the end to end delay.

fully preemptive operating system. With the Solaris 2 operating system, such a scheduling class exists and is called real-time (RT). Nevertheless, even when they run with the RT priorities, the processes only own a few real-time characteristics: their essential feature is that their priority class is greater than the one of the system tasks. On the other hand, as using the RT scheduling class is able to disturb the operating system because system tasks are delayed when a RT process runs, RT processing must be kept short. For instance, if the workstation is overload by RT tasks, communications (in system class) will not be processed anymore. Furthermore, if a RT process makes a system call, it loses its RT feature, and enters the system scheduling class. As a consequence, the RT scheduling class is essential for respecting the temporal synchronization constraints, but it has to be handled carefully [OWEZARSKI 96b].

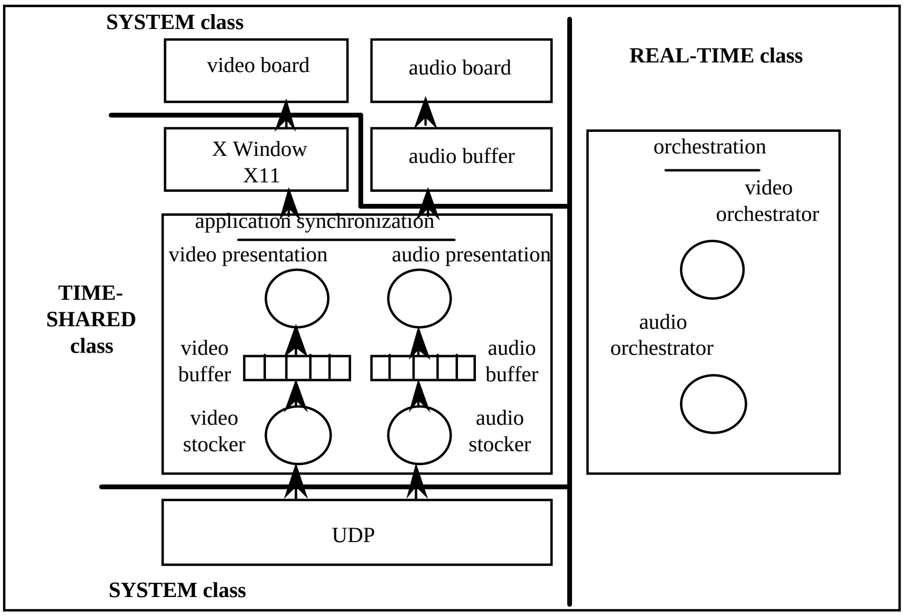

Figure 3. *Synchronization architecture for a visioconference application*

The architecture for the visoconference application resulting from the problems presented previously is depicted on Figure 3 [OWEZARSKI 95]. It depicts the different components of the visioconference system, associating to each of them their scheduling class in the operating system. In addition, this Figure shows the different tasks of the application, with:

- the audio and video stockers which get the data from the network, and store them in audio and video buffers. These audio and video buffers are used for temporarily storing the unsynchronized data coming from the network and to keep them long enough in order to solve the jitter problem;

- the presentation audio and video processes, running in the Time shared class, which perform the required operations and computing required for the sound and picture presentations;
- the real-time orchestration processes which play the synchronization scenario modeled by the receiver applicative TSPN and which control the presentation processes to guarantee the presentation temporal requirements.

With these processes, the intra-stream synchronization principle relies on controlling in real time the presentation processes thanks to the orchestration processes. The main idea deals with de-correlating presentation tasks suffering from the system asynchronism, from temporal control tasks. Inter-streams synchronization is implemented thanks to a rendez-vous between audio and video orchestration tasks which respect to the "and-master" firing semantic of the TSPN depicted on Figure 1.

4.3. Using a partial order transport

4.3.1. Architecture analysis

The QoS management principles presented in this chapter have been implemented for the PNSVS visionconference application. Measurements of audio and video objects presentation times for the application showed that synchronization constraints are perfectly respected [OWEZARSKI 98]. However, it appeared a problem of excessive losses when an unreliable network is used. In fact, each network loss infers more than a single discontinuity at the presentation interface level. Indeed, if an object is lost by the network and if this loss is acceptable according to the QoS requested by the user, then this loss lead to the duplication (at the presentation level) of the preceding object, what represents to a discontinuity (what seems to be the normal and minimal degradation in such a case). However, as PNSVS presentation processes [OWEZARSKI 96b] cannot determine whether this object has been lost or only delayed, they wait for it as long as possible (till the maximum presentation time of the preceding object expires) before starting an exception computing (replacing the missing object by another temporarily equivalent). Thus, in case of one loss, the application was unable to recover from it, and in addition, it wasted time waiting for the object to compute. Because of this time waste, or the accumulation of such time wastes, the end to end delay increases what can make the mechanism of delay control work, and create new losses. Similarly, the delay induced by waiting for an object makes the inter-streams drift increase, and when the inter-streams transition is fired, it can lead to an acceleration of the late stream, and thus causing new losses on this stream. The loss of one image which should only provoke one discontinuity can then have much more serious consequences.

To solve these problems, it is necessary to use a transport mechanism which delivers data and detect losses as early as possible.

4.3.2. A solution based on a partial order transport

[AMER 94] [CHASSOT 95A] [CHASSOT 95B] [DIAZ 95] define a new partial order transport protocol as a transport aiming at delivering objects sent on one or several connections, according to a given order. This order is any order between the total order (TCP) and no order (UDP), and can be expressed as a serial/parallel composition of objects. It appears that this order can be the one described by the constraints of the applicative TSPN [DIAZ 94B]. Thus, [AMER 94] [CHASSOT 95A] [CHASSOT 95B] [DIAZ 95] define this delivery according to a given order as a logical synchronization of multimedia objects.

In addition, this new notion of partial order is enriched by the partial reliability notion.

According to the issues encountered with the first version of PNSVS presented in previous section, the partial reliability notion is essential. This notion is tightly related to an end to end transport QoS that defines a nominal QoS and a minimal QoS under which the user requested service is not ensured. In term of reliability, this minimal QoS can be expressed in different ways: by a maximum number of losses inside a sequence, and/or by a maximum of consecutive losses. Thus, in case of an acceptable loss, detected when the received object is logically after the one expected, the object initially expected can be declared lost as immediately (no recovery attempt is initiated), and the object just received by the receiving transport entity is delivered to the application (earliest delivery). On the other side, if the loss is not acceptable according to the requested reliability, retransmission occurs, the number of retransmissions being a parameter of the transport service. Any object received by the receiving transport entity is then delivered as early as possible in respect with the partial reliability; if this object cannot be delivered according to the partial order, it is certainly because a network problem disturbed the transmission of objects which are logically before this later. Then, if the objects which precede logically the received one can be lost according to the partial reliability, they will be considered as lost (early losses), and the received object is not delayed anymore (early delivery) [AMER 93A] [AMER 93B].

In fact, two approaches exist for managing partial reliability: media per media and by group of media. In the media per media management, the receiving entity can only use partial reliability mechanisms on the stream it manages, and no on the other streams of the multimedia connection. On the other hand, with a by group of media management, the receiving entity of a stream can, for implementing the early delivery principle, declare some losses on other streams of the multimedia connection when there are inter-streams logical synchronizations [DIAZ 95].

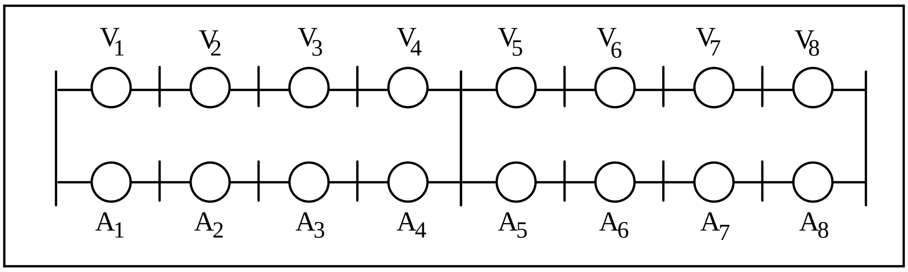

Figure 4. *Partial order example*

Let us consider the example of Figure 4 that represents one Petri Net of a serial / parallel composition for a multimedia connection (audio and video). Let us assume that the maximum number of losses on a stream for each inter-stream synchronization period is one object. Let us now assume that objects V1, A1, A2, and A3 have been received by the receiving transport entity and delivered to the application. Let us assume now that the receiving transport entity receives the V4 object; by respecting the partial order on the video connection, the V4 object cannot be delivered, as it would require to declare as lost objects V2 and V3. But two losses per synchronization period and per stream are forbidden. The V4 object is then stored, waiting for new object receptions. If object V3 arrives, it can be delivered if object V2 is declared as lost. In order not to delay transmissions, the partial order transport delivers as early as possible the received or stored objects; it then declares the loss of object V2 (as soon as it receives V3), and delivers sequentially V3, and V4 which was previously stored: principle of the earli delivery, permitted because of an early loss declaration. This illustrates the per media management mechanism of the partial reliability principle.

Let us now assume that the V5 object arrives. Regarding the partial order, this object cannot be delivered, because it is logically after the A4 object (because of the inter-streams synchronization after A4 and V4). With a per media management of the partial order multi-connection, the V5 object has to be stored as long as the A4 object arrives or is declared lost (because of the arrival of A5).

However, if we consider a per group of media management, the manager of the video connection can also declare some losses on the audio connection. In this case, as one loss is acceptable on the audio stream for the first period, it then declares the A4 object as lost in order to deliver V5 to the application as early as possible. This by group of media partial reliability management needs that the architecture of the receiving transport entity (Figure 5) integrates a manager for multimedia multi-connections [CHASSOT 95C].

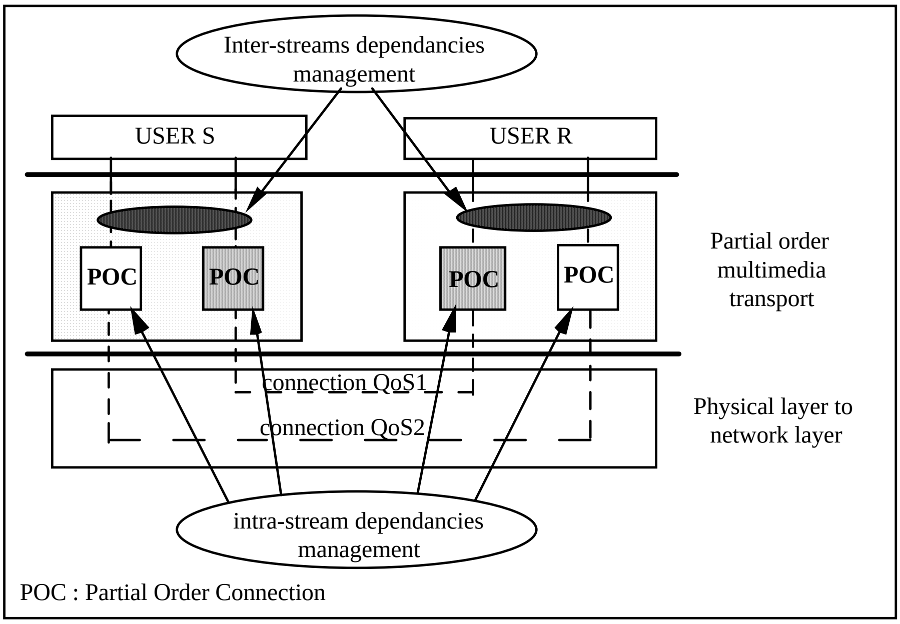

Figure 5. *Architecture of the sending and receiving entities of a partial order transport*

4.3.3. Architecture for an application over a partial order transport

The architecture required to run PNSNS on top of a partial order transport is depicted on Figure 6. This architecture does not directly put the synchronization task on top of the partial order transport: because of its logical management of data, the partial order transport cannot detect long losses sequences, and then does not provide, in this case, any improvement compared to a transport protocol as UDP. These losses in sequence cannot be detected without an explicit management of time (thanks to a real or relative clock). Thus, the pre-synchronization layer has been added between the transport and application layers for providing a temporal control on data delivered or lost by the transport; it allows the detection of long losses sequences, network drifts and communication system asynchronism issues.

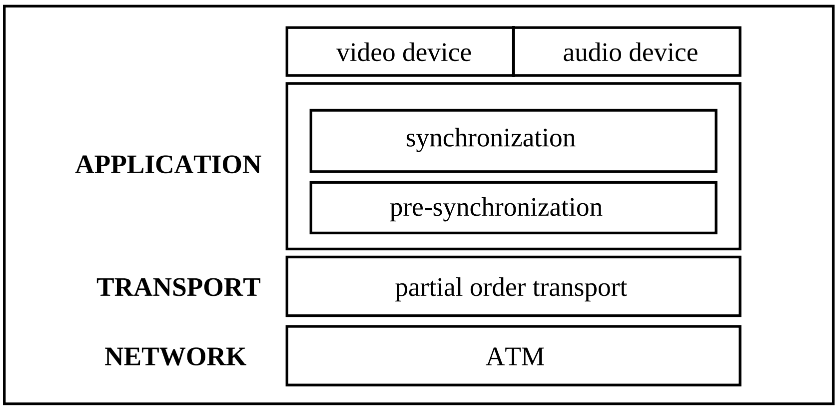

Figure 9. *PNSVS architecture on top of a partial order transport service*

The software architecture of this second version of PNSVS is very similar to its first version. In the second version, it only appears two pre-synchronization processes which control the temporal behavior of stockers, in the same way as orchestrators control the temporal behavior of presentation processes.

4.3.4. Modeling of the different synchronization levels

It is to note that the Petri Net model is used for modeling the behavior of each layer: user interface, synchronization application, transport, etc. and this with different behaviors for each layer. In fact, Figure 7 shows how it is possible to model, level by level, the behavior of each layer according to their functionalities and constraints they have to respect in this architecture (with 4 levels).

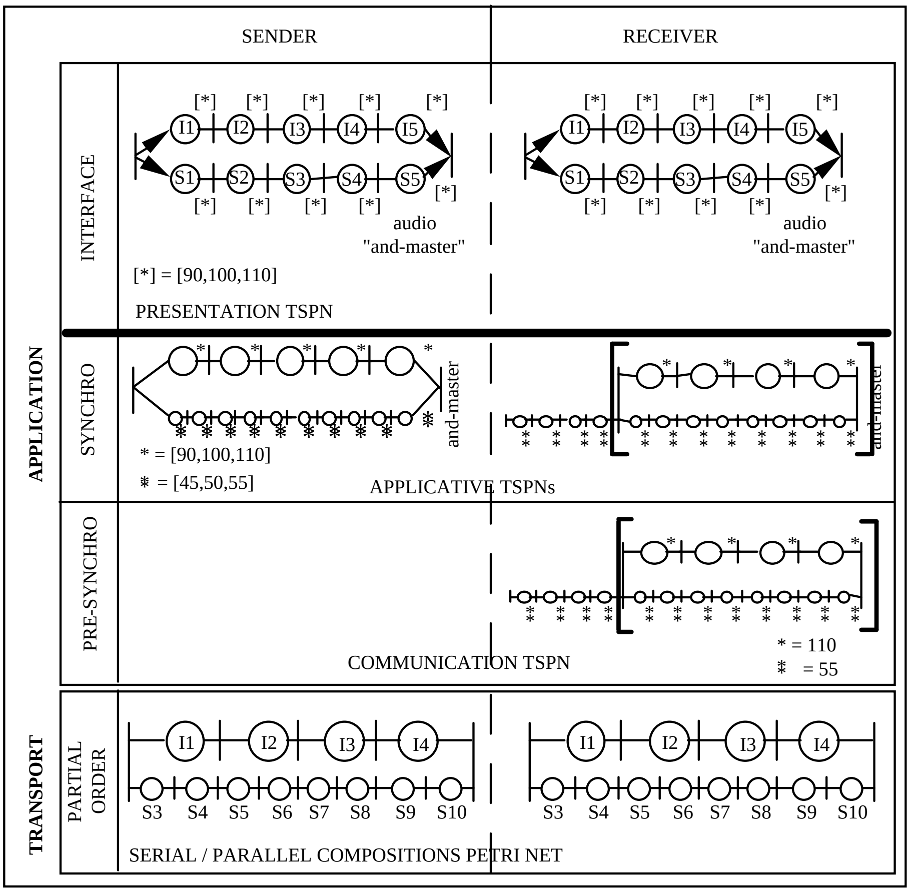

Figure 7. *Modeling of the behavior of each level of PNSVS*

4.3.5. Architecture evaluation

The PNSVS application has been implemented on Sun Workstations (Sun SparcStation 10, 5 or 2) with the Solaris 2.5 operating system. These machines were equipped with Parallax video boards which allow the capture, display, compression and uncompression of images at the M-JPEG format. The audio board is the standard one on these kinds of machines. Tests were performed on a 10 Mbps Ethernet and a 155 Mbps ATM networks.

PNSVS is a visioconference application which can handle 25 images/s (320 x 240 pixels and 24 bits coded colors) full duplex. The minimum end to end presentation delay obtained is around 400 ms and seems very difficult to reduce because of the audio board latency time (around 250 ms).

It has been shown that the temporal constraints of audio and video presentation (jitter and drift) are perfectly enforced [OWEZARSKI 98].

In addition, benefits due to the use of a partial order transport on an application as PNSVS have been evaluated. Thus, Figure 8 presents, according to a simulated network loss rate, the additional loss rate due to the first version of PNSVS (which takes advantage of UDP) and the second version of PNSVS (which takes advantage of a partial order transport). The curves exhibit a significant quality improvement when the partial order transport is used.

% of network losses / % of applicative losses	0	5	10	20	30	40	50	60
POC	2	0,4	0,1	0,9	1,7	2,3	2,9	3,7
UDP	2	2,4	3,7	5,5	7,4	9,1	10,9	12,7

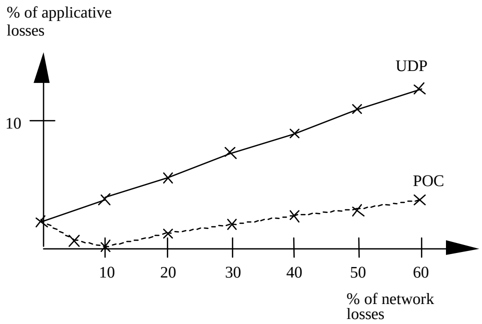

Figure 8. *Comparative assessment of losses with POC and UDP for PNSVS*

5. Conclusion

This chapter has presented a new and adequate architecture and its related mechanisms to fulfill important synchronization requirements for multimedia applications in asynchronous environments. To reach the best possible QoS, the synchronization architecture is limited by two extreme layers: an applicative synchronization layer that ensures the multimedia objects temporal requirements and a new multimedia advanced partial order transport layer. However, to interface the partial order transport service with the application needs, a pre-synchronization level has been located between the application and the transport levels. While conceptually this level should be located inside the transport layer, due to implementation constraints, it has been here situated in the application layer.

This architecture has been derived after considering a formal representation of the multimedia information [DIA97]. It has been shown that it is possible to model the behavior of all layers of such a synchronization architecture. The model that has been used here is a time Petri nets based model, the TSPN presented in chapter 6. A presentation TSPN, deduced from the user QoS (see section 3.2), is used to model the presentation level multimedia synchronization scenarios at the interface between the application and the user. Then, in the general case, a modified applicative representation (for the sender and the receiver) is deduced from it to model the synchronization application behavior. Finally, the receiver applicative model leads to the design of the partial order transport.

These concepts have been used to design the PNSVS videoconferencing system. Its implementation has been based on advanced system mechanisms, such as the real-time scheduling class of Solaris 2, as described in section 4.2.3, that perfectly fulfills the videoconference requirements. The new transport architecture, service and protocol based on partial orders, running on the Solaris 2 streams mechanisms, provides the basis of the global architecture. Then, an evaluation has shown the efficiency of the architecture compared to the traditional one based on UDP/IP.

References

[ADELBERG 94] ADELBERG B., GARCIA-MOLINA H., KAO B., "Emulating Soft Real Time Scheduling Using Traditional Operating System Schedulers", Proceedings of the Real Time Systems Symposium, San Juan, Puerto Rico, December 7-9, 1994

[AMER 93A] AMER P.D., CHASSOT C., CONNOLLY T., DIAZ M., "Partial order transport service for multimedia applications : reliable service", proceedings of the second High Performance Distributed Computing Conference, July 1993

[AMER 93B] AMER P.D., CHASSOT C., CONNOLLY T., DIAZ M., "Partial order transport service for multimedia applications : unreliable service", proceedings of the 3rd International Networking Conference, INET'93, August 1993

[AMER 94] AMER P.D., CHASSOT C., CONNOLLY T., CONRAD P., DIAZ M., "Partial order transport service for multimedia and other applications", IEEE/ACM transactions on Networking, Vol. 2, No. 5, October 1994

[BAKER 94] BAKER T.P., MUELLER F., RUSTAGI V., "Experience with a Prototype of the POSIX Minimal Realtime System Profile", IEEE Newsletter on Real Time Systems, 1994

[BLAKOWSKI 95] BLAKOWSKI G. et STEINMETZ R., "A media synchronization survey : reference model, specification and case studies", *IEEE journal on selected areas in communications*, Vol. 14, No. 1, January 1996.

[BLAKOWSKI 96] BLAKOWSKI G., STEINMETZ R., "A media synchronization survey: reference model, specification, and case studies", IEEE journal on selected areas in communications, Vol. 14, No. 1, January 1996

[CHASSOT 95A] CHASSOT C., DIAZ M., LOZES A., "From the partial order concept to partial order multimedia connections", Journal of high speed network (JHSN), 1995

[CHASSOT 95B] CHASSOT C., DIAZ M., LOZES A., "Principes d'implantation d'une connexion multimédia d'ordre partiel", 4ème colloque francophone sur l'ingéniérie des protocoles, CFIP'95, Rennes, France, 10-12 mai 1995

[CHASSOT 95C] CHASSOT C., "Architecture de transport multimédia à connexions d'ordre partiel", Thèse de doctorat de l'institut National Polytechnique de Toulouse (INPT), Décembre 1995

[COULSON 94] COULSON G., BLAIR G.S., ROBIN P., "Micro-kernel support for continuous media in distributed systems", Computer Networks and ISDN systems, Vol. 26, 1994

[COURTIAT 96] COURTIAT J.P., DE OLIVEIRA R.C., "Proving temporal consistency in a new multimedia model", Proceedings of ACM multimedia'96, Boston, November 18-22, 1996

[DIAZ 93A] DIAZ M., SÉNAC P., "Time stream Petri nets, a model for multimedia streams synchronization", proceedings of MultiMedia modelling'93, Singapore, November 1993

[DIAZ 93B] DIAZ M., SENAC P., DE SAQUI-SANNES P., "Un modèle formel pour la spécification de la synchronisation multimédia en environnement distribué", Actes du colloque francophone sur l'ingénierie des protocoles, Montréal, Canada, 7-9 septembre 1993

[DIAZ 94A] DIAZ M., OWEZARSKI P., "Développement d'un système de visioconférence sur réseau local", Rapport LAAS No. 94354, Juillet 1994

[DIAZ 94B] DIAZ M., LOZES A., CHASSOT C., AMER P.D., "Partial order connections : A new concept for high speed and multimedia services and protocols", Annals of telecommunications, Vol. 49, No. 5-6, May-June, 1994

[DIAZ 95] DIAZ M., DRIRA K., LOZES A., CHASSOT C., "Definition and representation of the quality of service for multimedia systems", 6th International conference on high speed Networking, HPN'95, Palma de Mallorca (Balearic Islands), Spain, September 11-15, 1995

[ISO 90] ISO-IEC JTC1/SC2/WG12, "Coded representation of multimedia and hypermedia information", Multimedia and hypermedia information coding expert group (MHEG), working document version 2, July 1990

[ISO 93] ISO, "Coded representation of multimedia and hypermedia information objects (MHEG)", Committee draft, ISO/IEC CD 13522-1, June 1993

[JEFFAY 92] JEFFAY K., STONE D.L., DONELSON SMITH F., "Kernel support for live digital audio and video", Computer communications, Vol. 15, No. 6, July / August 1992

[JEFFAY 94A] JEFFAY K., STONE D.L., DONELSON SMITH F., "Transport and display mechanisms for multimedia conferencing across packet-switched networks", Computer networks and ISDN systems, Vol. 26, 1994

[JEFFAY 94B] JEFFAY K., "On latency management in Time-Shared Operating Systems", IEEE Newsletter on real time systems, 1994

[KANG 94] KANG G. SHIN, PARAMESWARAN RAMANATHAN, "Real time computing : a new discipline of computer science engineering", Proceedings of the IEEE, Vol. 82, No. 1, January 1994

[KATCHER 94] KATCHER D.I., KETTLER K.A., STROSNIDER J.K., "Modeling DSP Operating Systems for Multimedia Applications", Proceedings of the Real Time Systems Symposium, San Juan, Puerto Rico, December 7-9, 1994

[LIOU 91] LIOU M., "Overview of the px64 kbits/s video coding standard", Communications of the ACM, Vol. 34, No. 4, April 1991

[LIT90] T.D.C. Little, A. Ghafoor, "Synchronization and storage models for multimedia objects", IEEE journal on selected areas in communications, Vol. 18, No. 3, pp 413-427, april 1990

[OWEZARSKI 95] OWEZARSKI P., DIAZ M. et SENAC P., "Modélisation et implémentation de mécanismes de synchronisation multimédia dans une application de visioconférence", Actes du colloque francophone sur l'ingéniérie des protocoles (CFIP'95), Rennes, France, 1995.

[OWEZARSKI 96a] OWEZARSKI P.ET DIAZ M., "Models for enforcing multimedia synchronization in visioconference applications", Proceedings of the 3rd MultiMedia Modeling conference – Towards the information superhighway (MMM'96), World Scientific editor, Toulouse, France, 1996.

[OWEZARSKI 96b] OWEZARSKI P., "Conception et formalisation d'une application de visioconférence coopérative. Application et extension pour la téléformation", thèse de doctorat, Université de Toulouse III, France, décembre, 1996.

[OWEZARSKI 98] OWEZARSKI P., DIAZ M. ET CHASSOT C., "A Time Efficient Architecture For Multimedia Applications", *IEEE Journal on Selected Areas in Communication JSAC.*, April 1998, vol.16, n°3, p. 383-396.

[ROTHERMEL 95] ROTHERMEL K., HELBIG T., "An Adaptative Stream Synchronization Protocol", proceedings of the Network and Operating System Support for Digital Audio and Video coference (NOSSDAV'95), 1995

[SÉNAC 94] SÉNAC P., DIAZ M., DE SAQUI-SANNES P., "Toward a formal specification of multimedia synchronization", Annals of telecommunications, May / June 1994

[TURLETTI 93] TURLETTI T., "H.261 software codec for videoconferencing over the internet", Rapport de recherche INRIA No. 1834, Janvier 1993

[VOGEL 95] VOGEL A., KERHERVÉ B., VON BOCHMANN G., GECSEI J., "Distributed Multimedia and QoS : A Survey", IEEE Multimedia, Summer 1995

[WALLACE 91] WALLACE G.K., "The JPEG picture compression standard", Communications of the ACM, Vol. 34, No. 4, April 1991