New architecture for enforcing multimedia synchronisation in videoconferencing applications

Philippe Owezarski, Michel Diaz

LAAS - CNRS
7 Avenue du Colonel Roche
F-31077 Toulouse cedex, France

This paper presents a formally defined architecture allowing the design and the corresponding implementation of a set of modules that fulfill the quality of service requirements of PNSVS (Petri Net Synchronised Videoconference System), an application designed to run on top of an asynchronous environment. It describes the mechanisms and architecture allowing the system to enforce the temporal synchronisation between the audio and video streams, the presentation quality, and the end to end delay. This synchronisation method relies on a formal model extending time Petri nets : the Time Stream Petri Net model (TSPN), that will be used to completely specify the time constraints of the application streams. From this, the communication architecture is improved by introducing a new partial order transport service that maximizes the PNSVS quality of service. The partial order transport principle is presented together with the pre-synchronisation sub-layer, which makes the partial order transport service match the applicative synchronisation requirements. Moreover, it will be shown that each layer of this synchronisation architecture can be formally modeled by a welle defined set of TSPN models.

1. Introduction

The essential characteristic of new multimedia data is the wide range of new QoS constraints that have to be respected by all distributed systems that have to support multimedia applications. Among the different new QoS requirements of multimedia data, the most important and difficult to handle are certainly the new temporal constraints inherent to the flows of dynamic media, as audio and video.

This paper, then, aims to study and develop a new communication architecture and a set of adequate mechanisms to ensure some well defined quality of service parameters like audio quality, video quality and rate, end to end delay to users. For this purpose, we first analyzed in depth the multimedia synchronisation constraints of visioconference applications. In this case, the problem of multimedia synchronisation consists in ensuring both the intra and inter streams temporal requirements. Ensuring the intra stream temporal constraints consists in controlling the jitter on each audio or video object in order to keep it under acceptable maximum values. Ensuring the inter streams synchronisation consists in controlling the drifts between the audio and video streams (due to the cumulative effects of jitters) within a maximum value (sounds to pictures in lip synchronisation for example).

The multimedia synchronisation problem is the most important and difficult to solve in the design of multimedia distributed applications and systems [BLA96]. In fact, it is difficult to solve because of the asynchronous nature of available wide area distributed systems, and also

because of the wide range of synchronisation requirements users and applications can require : it is then needed to precisely, i.e. formaly, express these synchronisation requirements. For modeling the user temporal synchronisation constraints the TSPN (Time Streams Petri Net) model has been used, because it completely and easily expresses multimedia synchronisation scenarios. Because of a number of difficulties due to asynchronous distributed systems, it will appear that the application behaviour cannot be modeled using the more general model called the presentation TSPN. It will be shown that another model - the applicative TSPN - taking into account the operating system characteristics, is required. Given this applicative TSPN, it is furthermore needed to infer an adequate temporal scheduling of the presentation and orchestration processes. This model has then been used for modeling the PNSVS (Petri Net Synchronised Videoconference System) application that has been chosen as the exemple for illustrating the synchronisation mechanisms and the architecture proposed in this paper.

This paper will also show how the PNSVS quality of service can be improved if it is supported by a new multimedia transport protocol, taking into account themultimedia streams QoS, and ensuring a well defined delivery of the visioconference data (logical synchronisation). Such a transport protocol is based on a partial order model, depending on the TSPN that represents the applicative synchronisation scenario. In particular, it will be shown that the logical order of objects delivery will be modeled by a Petri Net derived from the applicative TSPNs. It is interesting to note that all problems reported and solved in this paper have been observed on Sun Solaris 2.x workstations.

The remainder of this paper is as follows. First, the QoS parameters of PNSVS are presented in Section 2. Section 3 shows the difficulties met when using asynchronous distributed systems. Section 4 presents the presentation TSPN model and the way it is used in PNSVS. Given this model - the presentation TSPN - and the problems inherent to the implementation of synchronisation mechanisms in asynchronous systems, Section 5 shows how it is possible to ensure temporal constraints using some mechanisms of the Sun Solaris 2 operating system. A new model - the applicative TSPN - allowing to model the applicative behaviour of the synchronisation software is described. Section 6 shows how PNSVS can take advantage of a partial order transport to improve the applicative QoS, and how the requested transport behaviour can be modeled also using a Petri Net. This leads to a formal representation of the multimedia architecture from the application to the transport layer and derived following a top-down approach. Finally, Section 7 presents some synchronisation measurements performed on the PNSVS implementation. Some remarks and perspectives conclude this paper.

2. The visioconference QoS features

As previously introduced, an important characteristic of multimedia data is the wide range of QoS they require, in particular by integrating a temporal dimension, with tight synchronisation constraints. This section then presents the specific QoS parameters of a videoconferencing application that has been chosen as the case study for this paper.

The QoS parameters of a visioconference application can be classified into two groups, i.e. as static and dynamic.

The static parameters, like the image size or quality, or the audio stream quality, directly impact the amount of resources required by the application, but have no influence on the algorithms of the application.

The dynamic parameters impact the amount of resources needed, and also the application code because they force the programmer to integrate into the application specific mechanisms allowing the system to fulfil these parameters. The dynamic parameters used here belong to three families : the multimedia synchronisation parameters, the dynamic parameters of the presentation quality, and the presentation end to end delay.

• The multimedia synchronisation parameters define the intra and inter stream constraints of multimedia objects. These parameters are temporal : they define the maximum jitter acceptable on each object and the maximum drift that can appear between two streams.
• The (simple) presentation quality parameters are essentially static parameters (image size and quality, etc.). However the maximum admissible number of media discontinuities is also a dynamic parameter. In fact, a lost data causes a synchronisation loss that can be difficult to handle. To solve this problem the lost data is usually replaced by another data temporally close or equivalent (generally the preceding data is presented twice), but this nevertheless leads to a discontinuity in the video stream presentation. Discontinuities are hurtful for the QoS, and their number has to be reduced as much as possible.
• The end to end delay defines the time between the grabbing on the sending workstation and the display on the receiving one. This delay must not be greater than a given value to provide a good interactivity between the users, and it has to be controled.

All these dynamic QoS parameters have to be taken into account by the application that has to include mechanisms to respect them.

3. Asynchronous distributed systems

The study and methodology described in what follows are based on the use of asynchronous distributed systems because :
• 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 synchronisation mechanisms that can be generalised for a maximum number of applications on a maximum number of platforms.
• the solution of the multimedia synchronisation problem in a synchronous environment is a particular case of the one in asynchronous environment, where asynchronism is null, and where solutions are simpler.
• with the real-time scheduling classes in Unix (Solaris 2), it becomes possible to implement real time applications (this was not the case before) [COU94] [JEF92] [JEF94b] [VOG95]. However, the gap between the theory of real-time system and the reality of experimental real-time available systems is large [KAT94]. 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 synchronising processes ;
- the kernel has to be fully preemptive ;
- system tasks can disturb the temporal scheduling of applications ;
- etc. …

In fact, for a real-time behaviour 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 [BAK94]. However, this is incompatible with distributed multimedia applications that have to access multimedia and network boards. On the other ahand, using classical systems (UNIX or POSIX) is of high interest, because they are widely available, and they inply only a few more problems than using "pure" real-time systems [ADE94]. In addition, [KAN94] showed that high speed wide area synchronous systems (including communication supports and operating systems) are impossible to realize.

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.

4. The TSPN model

From these constraints, it follows that an "acceptable" variability has to be defined. This will be done by using the TSPN model.

4.1. Definition

To design guaranteedsynchronised 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 synchronisation properties on the multimedia objects themselves, a model allowing the author of a multimedia application to model the application synchronisation constraints is required. Several studies have already been realized in this domain, and some models have been proposed. In particular, some of these models use formal approaches based on time Petri nets. The OCPN (Object Composition Petri Net) model [LIT90] only considers nominal computing times, and do not address the computing times variability of asynchronous systems ; Merlin's TPN (Time Petri Net) model [BER91] does not allow an easy modeling of inter-streams synchronisation between parallel streams.

The limitations of these models led [DIA93] [DIA94] to propose a new model, called TSPN, providing good expression and modeling powers [DIA93]. 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.

The inter-streams temporal drifts can be controlled in a very precise way using 9 different inter-streams transition semantics and the position of each time interval on the arcs leaving the places (allowing to compute each stream apart from the others). Using these transition rules, it is possible to specify synchronisation mechanisms driven by the earliest stream ("or" synchronisation rules), the latest stream ("and" synchronisation rules) or by a given stream

("master" synchronisation rules). This synchronisation semantics defines the synchronisation instants from an arc statically or dynamically chosen. For more details, [SEN94] [SEN96] gives a formal definition of the model and the different firing semantics.

4.2. TSPN usage in a visioconference application

It is then proposed to use TSPN models to describe synchronisation constraints for the PNSVS videoconference application. In PNSVS, the video and audio streams to be synchronised will be described by a TSPN.

As an example, Figure 1 describes a videoconference model for which the QoS parameters are :

• throughput at 10 images per second : it follows that the nominal presentation time of a video object is 100 ms ;

• acceptable jitter on audio or video : 10 ms (in absolute value) [JEF94a]. It follows that the temporal validity intervals are [90, 100, 110] on one image or on one audio packet ;

• audio is the most important media : inter-stream synchronisation is of "and-master" type for the sound. In fact, the sound being more important than the video , it is defined as the master stream and its temporal constraints have to always be fulfiled. Furthermore, the "and" part of this "hand-master" rule tries as much as possible to wait and respect the constraints on the video stream to reduce its discontinuities ;

• the synchronisation quality : the inter-stream drift must be less than 100 ms [JEF94a], 100 ms being the limit under which the temporal gap between audio and video cannot be noticed. Then the inter-stream synchronisation period corresponds to the presentation of 5 images. In fact, the maximum drift on 5 objects is 50 ms per flow and the inter-streams drift between the two flows is then less than 100 ms.

It follows that the TSPN of Figure 1 allows the user (author) to precisely describe the presentation requirements of one period of the videoconference application.

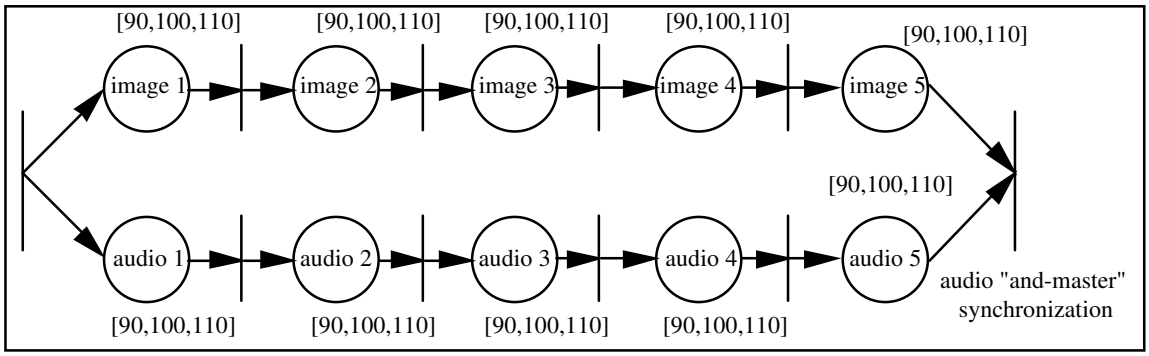

Figure 1. Visioconference TSPN example

5. Modeling and architecture of the PNSVS application

The aim of the synchronisation mechanisms and architecture is first to present the audio and video objects and second to fulfil their temporal synchronisation requirements. In fact, its goal consists in guaranteeing the QoS, i.e. the constraints defined by the presentation TSPN (Figure 1).

5.1. Multimedia boards latency and inter-stream synchonization : the drifted rendezvous

The presentation TSPN depicted on Figure 1 shows how units of the audio and video objects have to be synchronised. In particular, for a normal inter-stream synchronisation, any audio object i must be synchronised with image i.

Nevertheless, the multimedia boards do not have the same latency time : a video board can have a 50 ms latency time and an audio board a 250 ms latency time. Then if the application respects the presentation TSPN of Figure 1, the final presentation will not be synchronised : the audio part will be 200 ms late with respect to the video part, i.e. audio object i would be synchronised with image i+2, even if the synchronisation mechanisms have synchronised 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 rendezvous. In fact, the difference between the audio and the video board latency times being 200 ms, it is sufficient to synchronise the audio object i with the image i-2 : after having been handled by the presentation boards the audio object i will be synchronised with the image i. On this example, the control TSPN, here called the applicative TSPN, models the new synchronisation scenario of the application, with the drifted rendezvous. It is given on Figure 2.

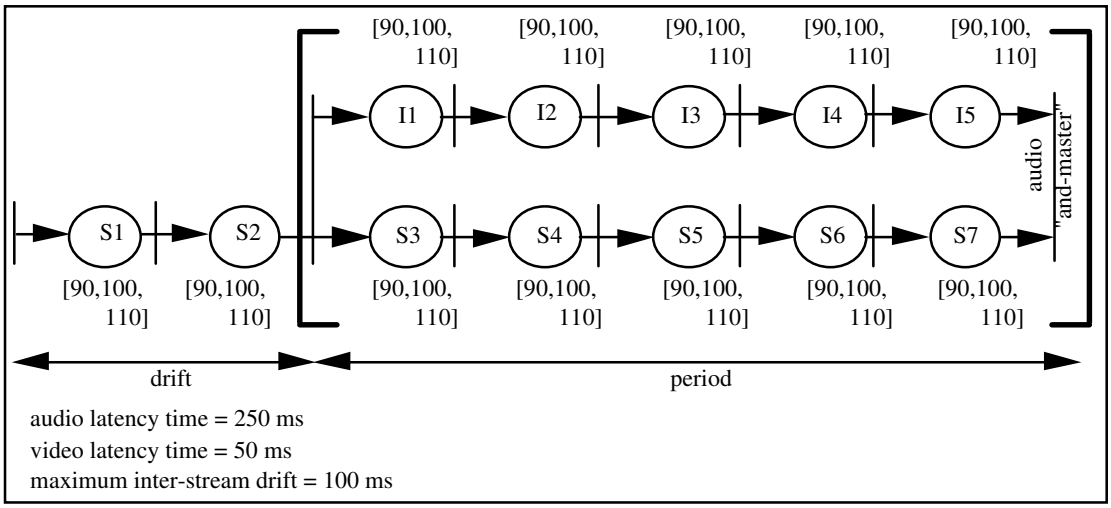

Figure 2. Applicative TSPN taking into account the hardware latency times

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, it remains 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 synchronisation period has to be changed. If the inter-stream synchronisation is enforced each 5 images, because of the 20 ms drift, 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. It follows that an inter-stream synchronisation must be done each 4 images, in order to make the inter-stream drift remain in the interval [-60 ms, 100 ms] (Figure 3).

The presentation TSPN modeling the multimedia synchronisation scenario and the applicative TSPN modeling the applicative processes behaviour have different shapes, because of the drifted rendezvous. Moreover, in the general case, the inter-stream synchronisation period has to be modified [OWE96a].

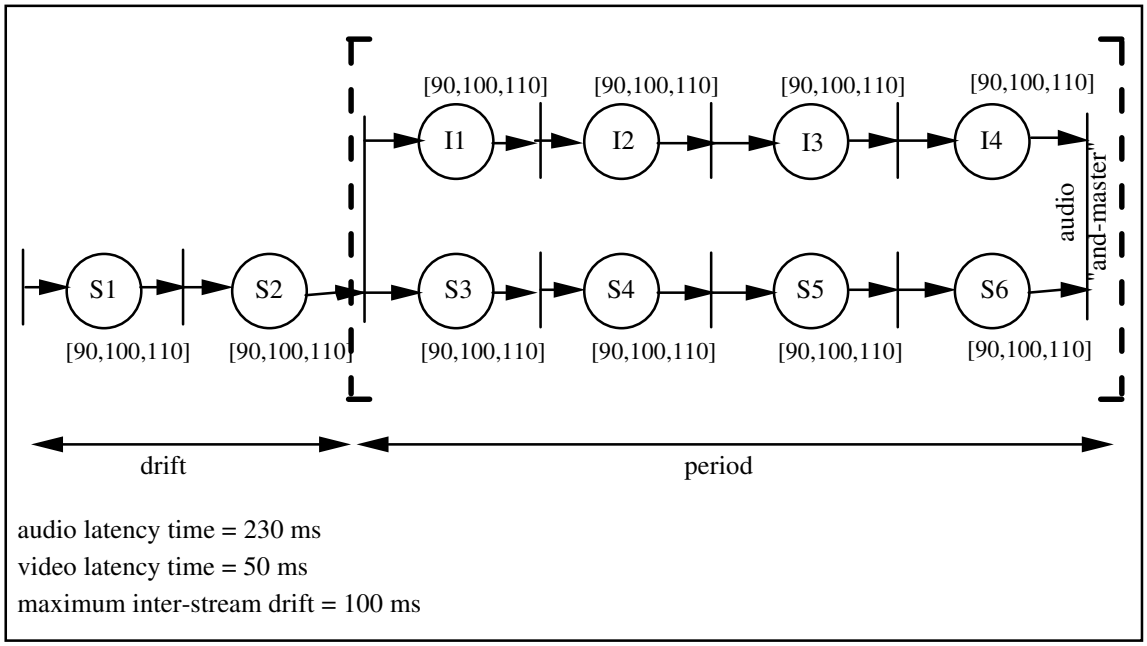

Figure 3. Applicative TSPN taking into account both the hardware latency times and the inter-stream synchronisation

5.2. End to end delay control

To solve the jitter problem, PNSVS stores incoming data in applicative buffers before their presentation. However, storing the PNSVS data increases the end to end presentation delay. This delay is the QoS parameter impacting the interactivity between the users : it implies the reception buffer sizes associated to each stream [JEF94a]. During the visioconference, the receiving application controls that the number of objects stored in buffers does not overrun the maximum bound in order to keep the end to end presentation delay under the requested value.

Nevertheless, if the receiving application becomes too delayed, it has to reduce its delay. The application knows it is late when the new incoming data cannot be stored anymore, and the delay has to be reduced by provoking the loss of the oldest data stored in the buffer (because its temporal semantic is no more valid).

The end to end presentation delay is quite hard to control, and it must be kept as short as possible. Studies [JEF94a] showed this delay must be under 250 ms. On some hardware and software it is impossible to respect this bound because of the boards latency times (400 ms latency time for the Sun audio board). Moreover, the receiving application has to store the data. In all cases, the computing times must be reduced to their minimal values.

• The multimedia boards latency times cannot be reduced by the application, as the communication system delays.

• The reception computing times cannot be substantially reduced. Images are atomic data : it is not possible to reduce the decompression / presentation time. For audio streams, the atomic element is a sample of 8 or 16 bits, and segmenting each samples packets does not significantly reduce the reception computing time.

• On the other side, it is possible to reduce the sender computing times : not the grabbing / compression time of an image, but the preparation time of an audio packet that depends on the packet size : in fact, the larger the packet is, the more time it requires to be produced (as samples are isochronously produced). For instance, on the applicative TSPN of Figure 2 to which an inter-stream synchronisation period modification has been applied, the 100 ms audio packets (800 bytes) requires 100 ms to be produced and 2 or 3 ms to be packetized and sent.

On the other side, as the computing time of one image (grabbing, compression, packetization and sending) requires only 55 ms (for a 320x240 pixels images, 24 bits of colors, and computed by a Parallax video board), the delay comes from the audio stream. As audio packets are non atomic, they can be segmented. Dividing the 100 ms audio packet by 2 makes the audio computing time be 52 ms, and the one on an image being 55 ms, it is possible to gain 50 ms on the end to end presentation delay. Note that as this delay is induced by the atomic video stream, it cannot be reduced anymore.

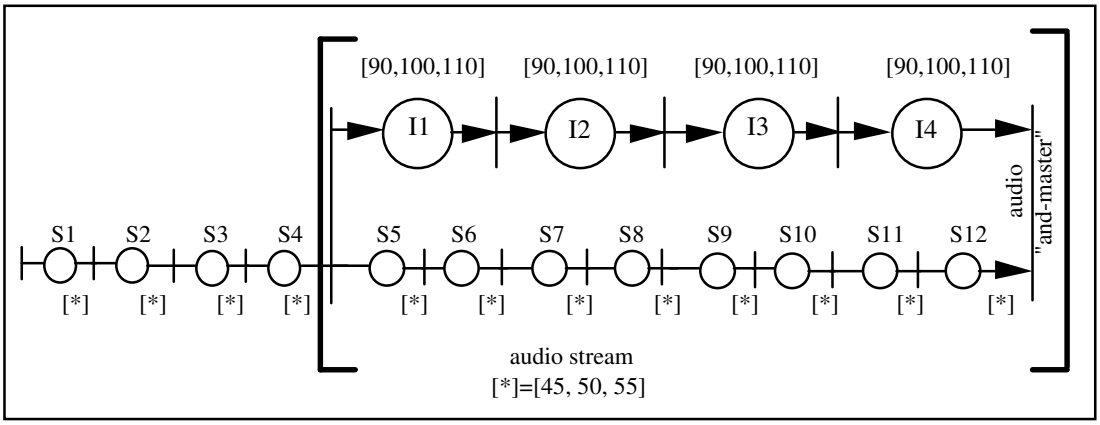

Figure 4. Applicative TSPN taking into account the end to end delay improvements

To preserve the logical synchronisation between the two streams, provoking the loss of one image induces the loss of all the audio packets associated to this image. Note that if a loss is provoked on the audio stream, it concerns all the audio packets associated to the corresponding image. For these reasons, the audio place number is a multiple of the video place number on a period. The applicative TSPN taking into account such characteristics and mechanisms is given on Figure 4.

5.3. Application architecture : Two level modeling for PNSVS

It now clearly appears that two models are needed to describe the two application levels. The first level is the interface modeled using the presentation TSPN. This TSPN models the application behaviour as seen from the users, i.e. how audio and video objects are synchronised 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. As already defined in Section 4, a presentation TSPN is a Petri net having temporal validity intervals on arcs leaving places, and typed firing rules on inter-stream transitions.

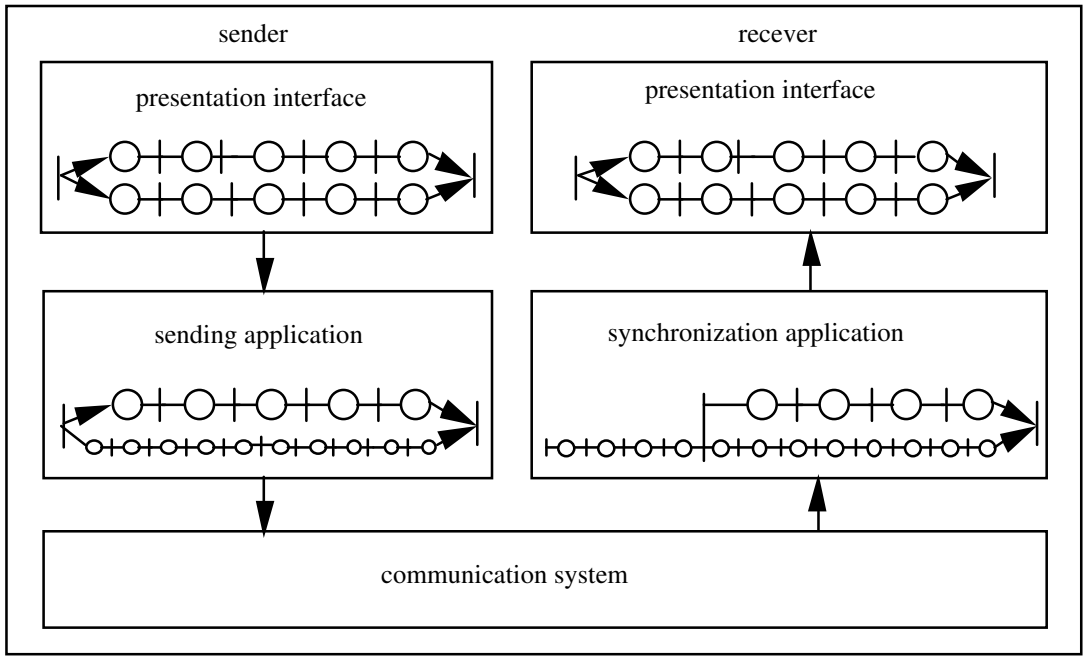

Figure 5. Two levels modeling

The second level is the applicative synchronisation. The applicative synchronisation can not be modeled using the presentation TSPN because of the characteristics of the operating system and of the multimedia boards. These characteristics force the synchronisation mechanisms to consider multimedia objects in a different way. To model the applicative synchronisation mechanisms, the applicative TSPN model has been defined. Moreover, the sender and receiver applications appear to be different, and two models are required. These two applicative TSPN have temporal validity intervals and inter-stream firing rules ; they also have parameters associated to places allowing one to parameterize synchronisation algorithms and calculate buffer sizes [OWE96b]. These applicative TSPNs that model the synchronisation and end to end control algorithms are deduced from the presentation TSPN, which is itself deduced from the QoS requested by users.

The two level hierarchy of the TSPN models is depicted on Figure 5.

5.4. Implementation of the mechanisms ensuring the synchronisation constraints

To fulfil the synchronisation requirements modeled by the presentation TSPN (Figure 1), the applicative synchronisation mechanisms have to implement the time relationships defined by the applicative TSPN (Figure 5). Because of the temporal validity intervals, this is equivalent to write a "weak synchronous" application.

In asynchronous distributed systems, as components are asynchronous (communication supports, operating systems, multimedia boards), synchronisation operations have to run at the highest level of the architecture, i.e. in the application layer of the receiver.

In fact, enforcing strong (final) synchronisation operations within the communication layers would often be useless because the data synchronised in the low layers can be desynchronised 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 synchronisation processes. In addition, the application layer (user part of the operating system) is the only layer where developers can change the processes scheduling. After having been handled by the synchronisation application, data are sent to the multimedia boards; they are handled again by the operating

system kernel, and its asynchronism that can desynchronise them. However, we can assume that the boards latency times are constant, or that the maximum jitter they imply is not sufficient to disturb the already enforced synchronisation. This hypothesis proved to be true on the tested hardware and software. Nevertheless, the most suited solution should be to have a single multimedia presentation board integrating in a synchronised way the audio and video computing. As long as such a solution will not exist, the correctness of the applicative synchronisation will depend on the truth of the (constant latency board) hypothesis.

Using this assumption for the multimedia presentation boards, the temporal synchronisation task is then reduced to implementing it at the application level. However, the application written in the user space of the operating system has to be weakly synchronous, and has to respect minimum and maximum presentation times. The operating system being asynchronous, as no bound on computing times is ensured in the time-shared scheduling, it is required to run processes which priorities greater than the ones of the system tasks, and to use a 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. 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 synchronisation constraints, but it has to be handled carefully.

The architecture of PNSVS resulting from the problems presented previously is depicted on Figure 6, where :

• the audio and video buffers receive the data from the network. These buffers are used to solve the jitter problem ;

• the presentation audio and video processes implement the required operations for the sound and picture presentations ;

• the real-time orchestration processes play the synchronisation scenario modeled by the TSPN and control the presentation processes to guarantee the presentation temporal requirements.

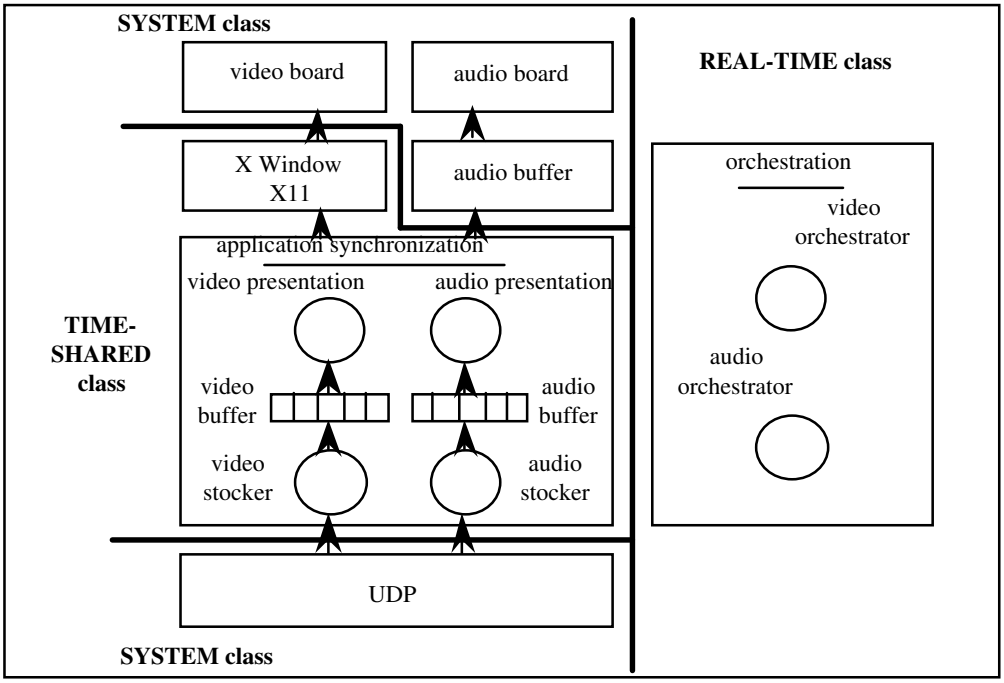

Figure 6. PNSVS synchronisation architecture

a) Intra-stream synchronisation

The intra-stream synchronisation algorithm (for the video and the audio streams) has to schedule 3 processes, implemented using threads. The proposed synchronisation approach gives a solution for the synchronisation problem in asynchronous environment, thanks to the use of the real-time scheduling class of the selected operating system (Solaris 2), and the use of a real-time timer ; this timer generates, when it times-out, an interruption at the greatest priority level, and is then not sensitive to asynchronism.

The intra-stream synchronisation algorithm, for video for instance, works as follows :

first, the video orchestrator creates the video presentation process while it sets the real-time timer ; the timer is planned to expire at the maximum presentation time of the current image minus the time required for the computing to perform in case of a timer expiration. This latest time is known and has no variability ; indeed, as the orchestration process runs in the RT scheduling class and do not perform any system call, operating system asynchronism is overcome (of course assuming that PNSVS is the only application using real-time processes on this machine).

Second, the presentation process performs the display operations and waits until the minimal presentation time is reached ; as the presentation process runs in time-shared class (TS) - or system class (SYS) when performing system calls - it supports the operating system asynchronism and its always positive jitters : the minimum presentation time is then ensured because of these positive jitters.

Finally, it remains to manage the completion of the presentation process. If it ends before its maximum presentation time, it informs the orchestration process and exits. If not, the timer expires, and the video orchestrator kills the presentation process to make it not last over the maximum presentation time. In this later case, as the orchestration process computing duration is known in advance and is taken into account by the real-time timer, the maximum presentation duration is respected.

After that, the video orchestrator has to start again the described process for the following image.

The presentation process principle is the same on the audio stream.

b) Inter-stream synchronisation

The inter-stream synchronisation algorithm is based on a rendezvous between the audio and video orchestration processes, this rendezvous having the semantic of the "and-master" firing rule. The principle of the temporal control is the same as the one described for intra-stream synchronisation, except that only the audio orchestration process uses a real-time timer (because only the presentation constraints on the audio object must be respected). The audio orchestration process has to wait until the end of the audio presentation process, and until the rendezvous call of the video orchestration process (meaning that the video presentation process is completed). Then, the inter-stream transition can be fired to start the next period. Nevertheless, if the real-time timer expires, the audio orchestration process has to kill the audio and/or video presentation processes and has to inform the video orchestration process to jump to the first object of the next synchronisation period. This case corresponds to a stream acceleration, and some images can be discarded.

6. Using a partial order transport in PNSVS

6.1. The problems

The main problem of the PNSVS version described in Section 5 is related to its dynamic quality of service. It is due to the use of a best effort transport like UDP : with such a transport protocol one object loss in the communication system can lead to many discontinuities in the stream presentation. It happens that if the communication service loses one object, the synchronisation mechanims 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, a discontinuity appears, what seems normal because a data has been lost. The problem is that the end to end delay also increased and other discontinuities can appear at the time of the "and-master" inter-stream transition firing (stream acceleration). Other discontinuities can also appear if the time lost during the waiting invokes the end to end delay control mechanism. The resulting strong quality degradation (several discontinuities) is due to the time waited when a data is lost : the application has to wait as long as defined by the maximum value of the TSPNs. This problem has no solution when a best effort transport protocol is used. The next part will show how a partial order transport protocol is able to solve it.

6.2. A partial order transport solution

A partial order transport service [AME94] [CHA95] [DIA95] is a transport which aims to deliver objects sent on one or several connections, following a given order. This order is any order between the total order (TCP) or no order (UDP), and can be expressed as a serial/parallel composition of objects (this order being for instance the one described by the constraints of the applicative TSPN). Thus, this delivery respecting a given order can be seen as a logical synchronisation of multimedia objects.

Furthermore, to solve the loss problem, the most interesting notion is the partial reliability notion. 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 loss number,… In the case where an acceptable loss can be detected (when a received object is not the one expected), the missing object can be declared lost as soon as possible (earliest detection of loss), and the next object already received can be delivered to the application without waiting (earliest delivery). If the loss is not acceptable in terms of requested reliability, retransmission occurs, the number of retransmission being a parameter of the transport service. The partial reliability (earliest losses and deliveries) defined in the partial order transport service are exactly deduced from the PNSVS requirements. The next Section proposes an architecture to adapt the transport service to the application requirements.

In fact, two approaches exist for managing partial reliability : media per media and by group of media. In the media per media, 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 can declare some losses on other streams of the same multimedia connection from the inter-streams logical synchronisations.

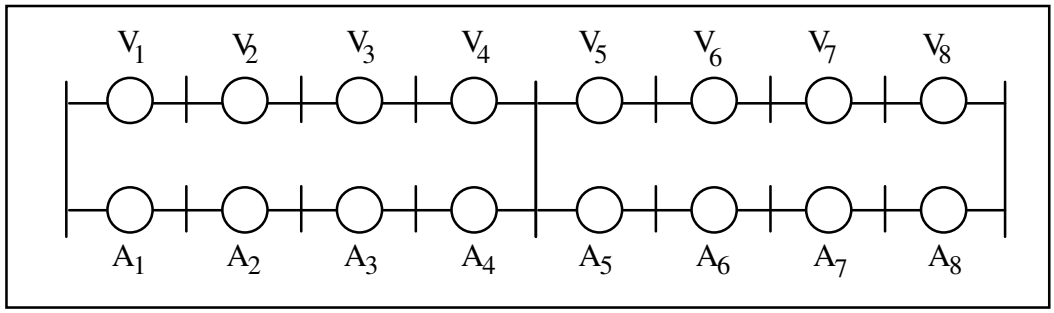

Figure 7. Partial order example

Let us consider the example of Figure 7 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 synchronisation period is of one object. Let us now assume that objects V1, V2, A1, A2, and A3 have been received by the transport receiving entity and delivered to the application. Let us assume now that the transport receiving entity receives the V4 object. From the partial order of the video connection, this object cannot be delivered to the application because it is logically located after the V3 object that has not been received. But, given the acceptable loss of one object V4 can be delivered if V3 is lost : this is the principle of the partial reliability. One loss being acceptable in each period for each connection, and in order not to delay the V4 object anymore, the V3 object will be declared lost, and the V4 object is delivered (principles of earliest loss and delivrance).

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 synchronisation 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 arriving 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 lost in order to deliver V5 to the application as early as possible. This by group of media partial reliability management leads to the design of the architecture of Figure 8 that integrates a manager for multimedia multi-connections [OWE98].

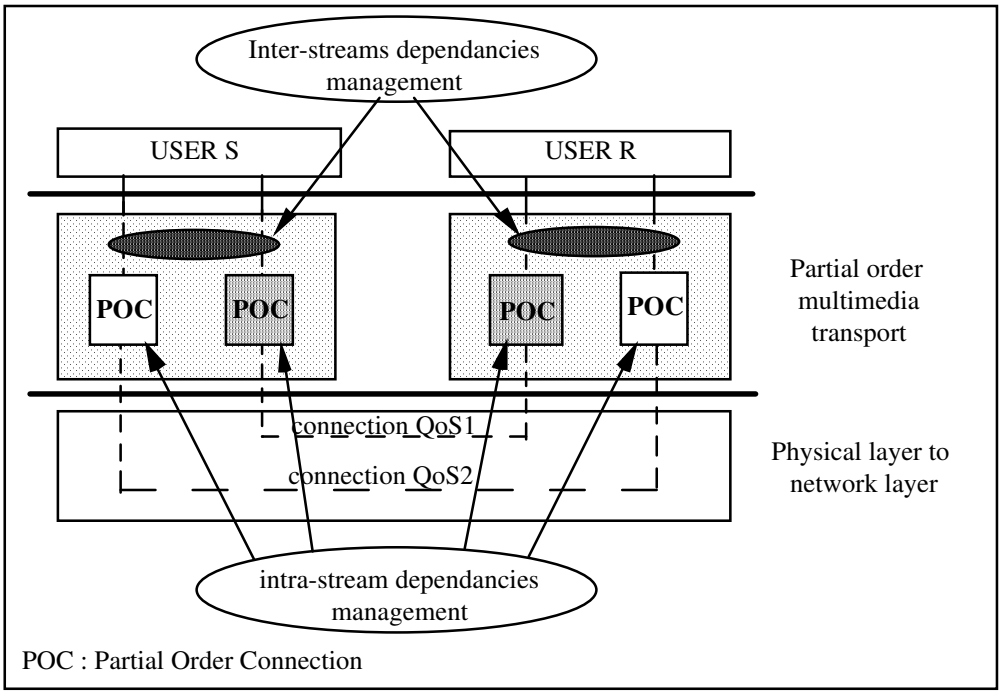

Figure 8. Architecture of the sending and receiving entities of a partial order transport

The following part presents an architecture for multimedia applications that takes advantage of a partial order protocol.

6.3. The PNSVS architecture using a partial order transport

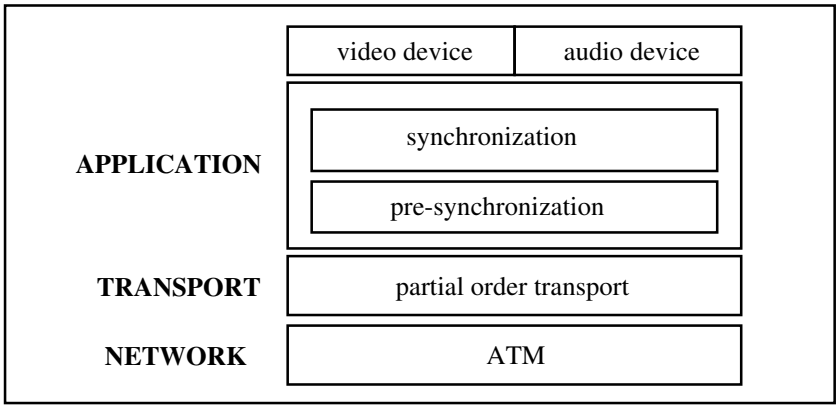

Figure 9. New PNSVS architecture

Because of the operating system asynchronism, the required architecture to run PNSVS over a partial order transport integrates a pre-synchronisation layer between the synchronisation and transport layers. Its purpose is to handle the temporal jitter of the partial order transport service. Figures 9 and 10 give its different building blocks.

This architecture has 3 functional levels : the partial order transport, the pre-synchronisation module and the synchronisation sub-layers.

• The transport layer has to create the high speed connections needed for the multimedia objects of PNSVS. Each of these connections has its own QoS in terms of throughput and reliability and, thanks to the partial reliability principles provides earliest losses and

deliveries, and then is as fast as possible. Nevertheless, the present partial order transport does not guarantee any temporal constraint on the isochronous streams it transmits, as it does not manage the time in an explicit way, but by using the transport buffers. The receiving entity of the partial order transport determines if it has to ask the sender for retransmissions, and the number of retransmissions it can ask for, in order not to overrun the maximum end to end presentation delay. This implicit time management is not precise. In ant case, because of the asynchronous operating system, an explicit time management cannot be precise enough at this level (in the SYS scheduling class), and such an explicit time management in the transport layer would increase uselessly the system load : explicit time should not be handled at this level.

• Because of this implicit time management by the transport buffers, the partial order transport protocol can only detect long loss sequences very lately, as these serial losses cannot be detected without an explicit time management (using a temporal clock). Thus, the pre-synchronisation layer has been added between the transport and the application in order to improve the temporal service provided by the partial order transport, and then to allow the improvement of the presentation quality of PNSVS. The pre-synchronisation layer losely manages the time in an explicit way, and ensures a temporal control of the delivered or lost data ; it allows to detect long loss sequences, network drift and asynchronous communication problems. It does not provide temporal gains as high as the earliest deliveries and losses of the transport service, but avoid internal time losses when some phenomenon disturbs the normal behaviour of the transport service. This temporal improvement relies on the same principle than the earliest loss handling, except that a decision is taken depending on the value of a clock instead on a logical order. The pre-synchronisation sub-layer is also responsible for the logical inter-stream synchronisation when losses are provoked. Note that this sub-layer exists only on the receiving machine : the sending machine has an empty pre-synchronisation sub-layer.

• The synchronisation sub-layer is not changed, as its mechanisms respect the behaviour modeled by the TSPNs. If the presentation and orchestration processes are not modified, the goal of the partial order transport and pre-synchronisation sub-layer is to avoid exception handling (as timer expiration) to take place. However, this synchronisation functions cannot be suppressed because partial order and pre-synchronisation run respectively in the system and time-shared scheduling classes, and are not able to guarantee temporal constraints.

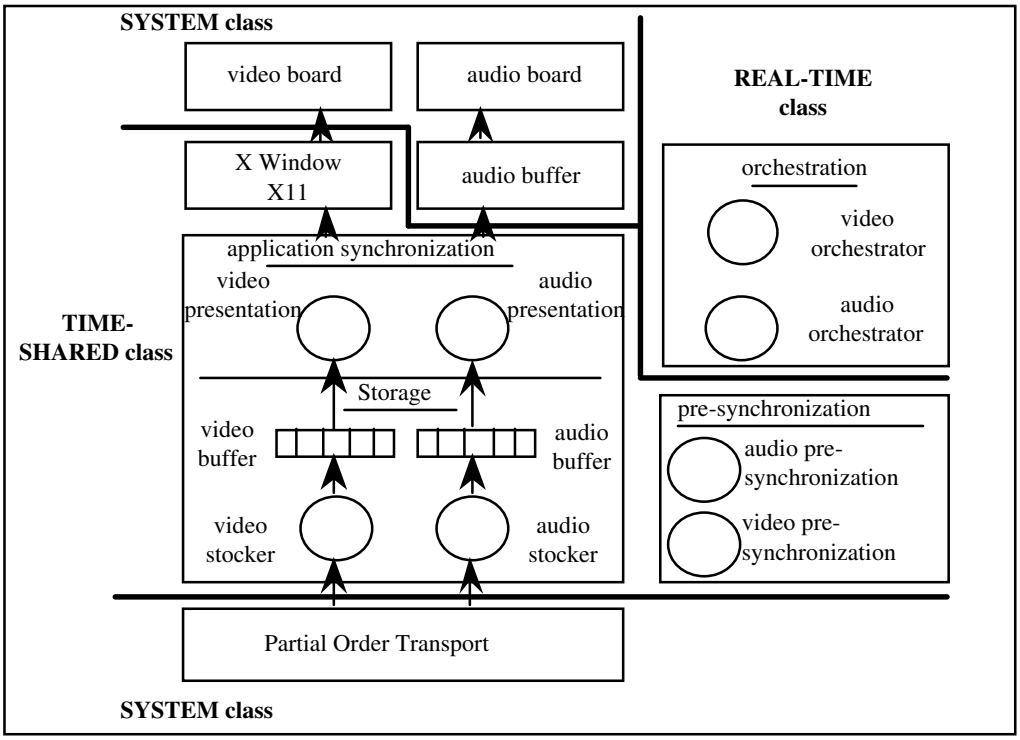

Figure 10. PNSVS Synchronisation architecture over a partial order transport layer

Benefits of the architecture

Because of asynchronous systems, temporal synchronisation is performed at the top level of the application and some real-time provisions have to be used. Such an approach allows three major improvements : (1) the too delayed data is discarded and an exception handling is started as soon as possible ; (2) processing is performed once and as early as possible (i.e. such that their effects will not be annihilated at the higher levels), in order to minimise the waiting time and maximise the QoS ; (3) storage is reduced : only the pre-synchronisation layer manages buffers in the user space.

6.4. Modeling the different synchronisation levels

Given this new extended synchronisation architecture including a pre-synchronisation sub-layer and a partial order transport layer, the model of the TSPN architecture given in Figure 5 can be extended in order to take into account the functionalities of these new layers.

The pre-synchronisation sub-layer has to detect late or lost data and to control end-to end delay. To model the pre-synchronisation sub-layer, a new TSPN is required : the communication TSPN. This new TSPN has the same shape as the receiving applicative TSPN, and contains the required parameters to be able to detect the late data and to control the end to end delay (all these parameters are associated to the places of the communication TSPN).

To model the partial order transport, it is required to express the partial order of the transport layer and its minimum QoS in terms of reliability. To model the partial order constraints, a partial order Petri net, deduced from the communication TSPN, is used. The reliability constraints are associated to the connections fdepending on the objects, in the same way as the required throughput.

Figure 11 shows the global example of the PNSVS modeling with the presentation, applicative, communication TSPNs together with the Petri net of the transport layer. However, for readability reasons, all parameters are not given in the Figure.

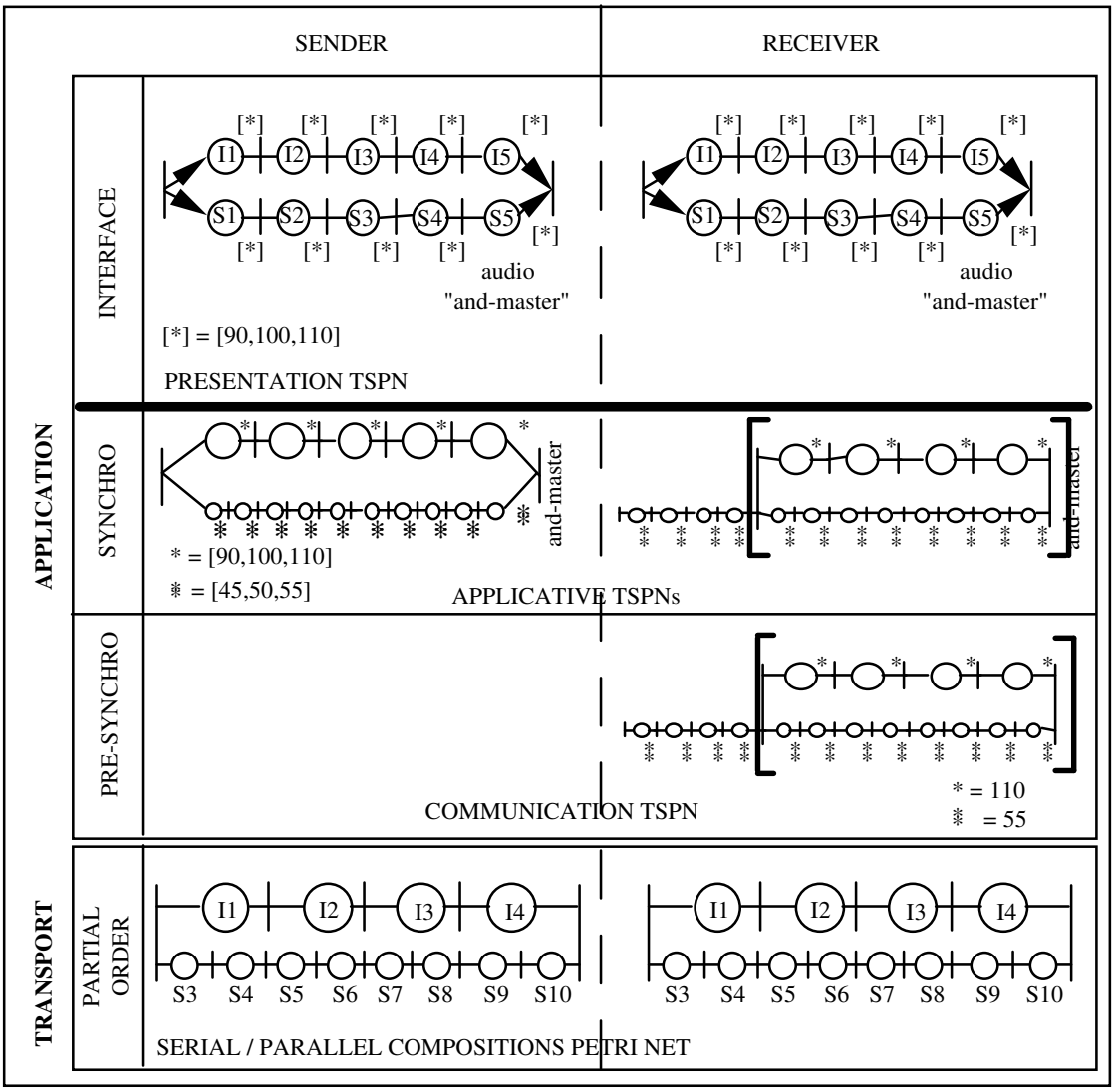

Figure 11. TSPNs architecture

7. Performance measurements

7.1. Synchronisation mechanisms evaluation

The PNSVS application has been implemented on Sun Workstations (Sun SparcStation 10) with Parallax video boards, and using an Ethernet and a 155 Mbits/s ATM network. This visioconferencing tool allows the users to create sessions with a 20 images / s video rate (320 x 240 pixels and 24 bits coded colors), a 64 kbits/s audio stream, and perfectly respected synchronisation constraints. 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).

This videoconferencing application has also been tested on an ATM WAN called MIRIHADE / SAFIR. MIRIHADE is the old name, SAFIR the new one of a long distance experimentation of ATM network in France. MIRIHADE / SAFIR is a french ATM network interconnecting several research centers in telecommunications (LAAS, INRIA, LIP6, etc.) in the same way as vBNS does in the US. MIRIHADE / SAFIR is devoted to the design and experimentation of new communication protocols and new distributed multimedia cooperative

applications issued from french research results. This ATM platform, separated from the global Internet, offers a 10 Mbps capacity between Paris, Toulouse, Nice, Rouen, Lyon and Grenoble.

It is important to test wether all selected synchronisation mechanisms fulfill the temporal presentation requirements, and in particular the quality of the intra and inter-streams synchronisation. The measurements have been realised in the case of a 10 images/s videoconference application where synchronisation requirements are modeled by the presentation TSPN of Figure 1.

Figure 12 shows, for the 100 first audio and video objects, the jitter that appears on the presentation of each. The measured jitter is the difference between the effective presentation duration and the nominal presentation time. For the video, the maximum jitter of 10 ms is never overstep, but the jitter is always negative ; in fact, in this experiment, there is no network problem (no loss and jitter), and data is available when the presentation process needs it. Thus the anticipation mechanism, that makes possible to stop the presentation of an object as soon as its minimal presentation time has been reached, always works. Some variations exist, less than 10 ms, due to the real-time scheduler of Solaris 2.

For audio, as for video, the intra-stream synchronisation requirements are always respected, but the jitter is always positive. In fact in this case, the firing of the intra-stream transition is controlled by the audio port signal. With a time scale expressed in millisecond, this consumption is always equal to 100 ms : variations are due to the time required by the system to take into account this information.

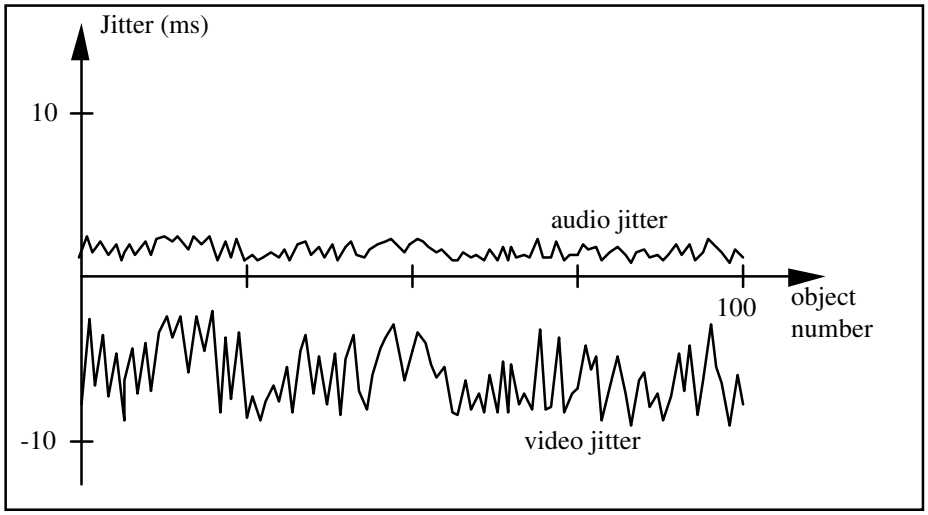

Figure 12. Jitter measurements for PNSVS

Figure 13 shows the curve representing the inter-stream drift for the audio and video sequence numbers. The video jitter is always negative and the audio jitter is always positive ; thus, the inter-streams drift (difference between the audio and video objects presentation dates) is positive. This drift increases during each period of 5 objects (due to the cumulative effect of jitter) and it is eliminated at each inter-stream synchronisation. The inter-streams synchronisation requirements are perfectly respected because the drift never exceeds 50 ms, the maximum allowed value being less than 100 ms.

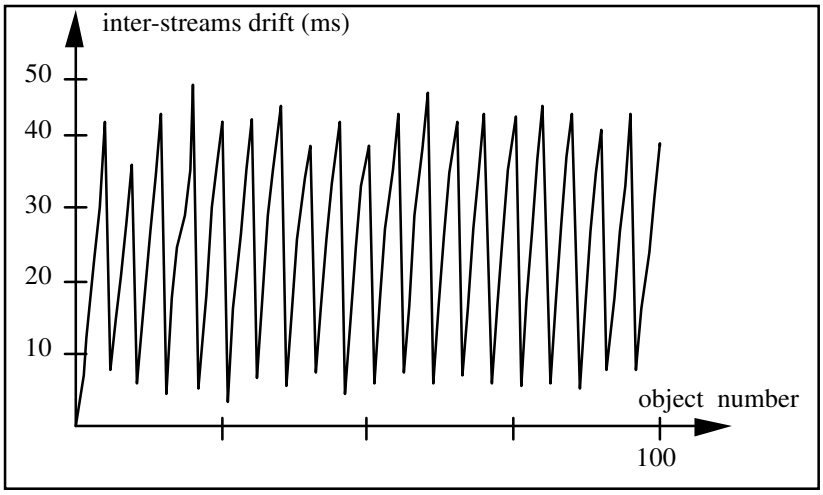

Figure 13. Inter-streams drift measurements in PNSVS

7.2. Overall architecture benefits evaluation

The partial order transport has been used to improve the PNSVS videoconferencing system. The new transport architecture, service and protocol based on partial orders, has been implemented using the Solaris 2 streams mechanisms, and it provides the basis of the global architecture.

It remains to evaluate the benefits of the overall architecture that includes the Partial Order Transport, the pre-synchronisation and the temporal synchronisation layers. As previously said, the use of a partial order transport (because of its earliest deliveries and losses mechanisms) avoids losing time at the application level. These time losses are really hurtful for the QoS because the application has to discard other presentation objects (images or sounds) when it has not enough time to compute them (end to end delay control mechanisms, "and-master" inter-streams synchronisation schemes, etc.).

Thus, the losses created by the synchronisation application will be measured in two cases : (1) when a partial order transport and (2) when a classical connectionless transport service (UDP) are used. To perform this evaluation, a network simulator allowing to simulate losses on the network has been used. This simulator is a Solaris 2 stream module, put on the sending machine (between the sending process and the ATM driver) that discards some packets. The PNSVS application evaluated is the one whose temporal constraints are expressed by the TSPN of Figure 1. The other parameters are : the maximum end to end delay 1 s, the audio latency 400 ms, and the video latency : 50 ms. With these values, the audio and video buffers can contain respectively 5 audio frames and 10 images. Results of measurements are described on Figure 14. It shows the applicative loss average in the 2 cases (using UDP and POC) depending on the network loss level.

Figure 14 shows that the QoS obtained by using POC is better than the one using UDP. In fact, when the application receives a loss indication from the partial order transport, it does not wait, and fires the TSPN transition as soon as possible (when the minimum presentation time is reached), while using UDP it has to wait the expiration of the timer associated to the maximum presentation duration. Using POC, each loss detection allows the application to "recover" the maximum allowed negative jitter ($n^s - x^s$), while using UDP it provokes a new time loss (equal to $y^s - n^s$).

Thus, the curve presenting the results of PNSVS using UDP grows linearly. The more network losses, the more time losses appear, and the more the application has to discard data to reduce the end to end delay.
Four comments can be made about the POC curve :
1) if the network is reliable, results using POC or UDP are equivalent ;
2) the applicative loss level decreases between 0 and 10 % of network losses. To explain this phenomenon, let us remember the results on audio and video jitter that are respectively 1 ms and -4 ms (on average). With such jitter values and a network loss level less than 10 %, the drift in a synchronisation period is positive (there is not enough network losses to make the application recover from the audio drift), and some applicative losses are created by the end to end delay control mechanism ;
3) the applicative loss level grows between 10 and 50 % of network losses. In this case, there are enough network losses to make the drift on a synchronisation period negative. However losses appear because of a starvation problem : each time a network loss appears, using POC, the end to end delay decreases, and this progressively frees buffers. When buffers are empty, even if the application is waiting as much as possible for the next object, this will not arrive before the application replaces it by a substitute (the last image for video, or an empty sound for audio stream) ;
4) finally, if the network loss level is greater than 50 %, long sequences (with more than 5 objects consecutively lost, 5 being the number of audio packets that can be stored in audio buffer) can appear. In this case, POC cannot detect all losses early enough, and the pre-synchronisation layer detects them. The principle of the pre-synchronisation mechanism is the same as the applicative synchronisation one. Pre-synchronisation is nevertheless more effective because it computes data earlier than the synchronisation layer. However, the drift on a synchronisation period can be positive and applicative losses are then caused by the end to end delay control scheme.

Another test for the effect of network jitter on PNSVS has also been made (using a network simulator). This test proves that PNSVS can always recover from jitter less than 100 ms. This result is quite interesting because, using ATM networks, such a value of the jitter jitter is a not reachable upper bound.

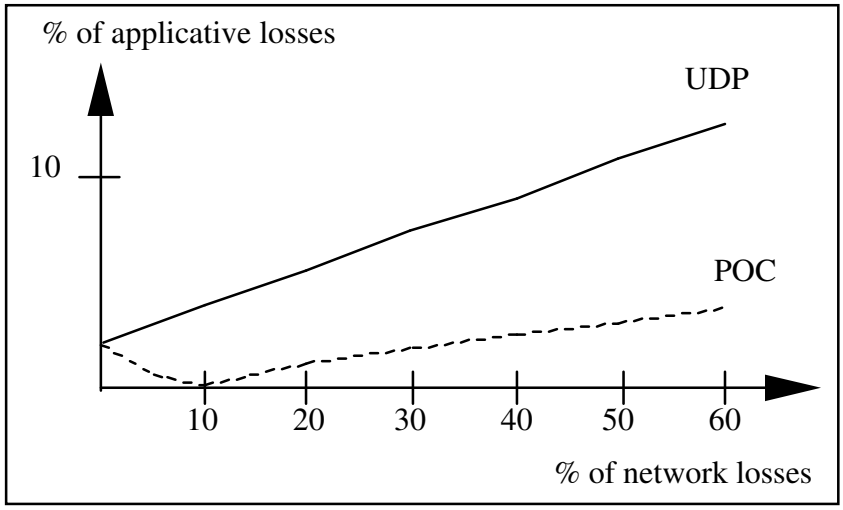

Figure 14. Overall architecture benefits evaluation

8. Conclusion

This paper has presented a new and adequate architecture and its related mechanisms to fulfil important synchronisation requirements for multimedia applications in asynchronous environments. To reach the best possible QoS, the synchronisation architecture is limited by two extreme layers : an applicative synchronisation 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-synchronisation 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 behaviour of all layers of such a synchronisation architecture. The model that has been used here is a time Petri nets based model, the TSPN presented in section 4. A presentation TSPN, deduced from the user QoS (see section 4.2), is used to model the presentation level multimedia synchronisation 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 synchronisation application behaviour. 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, and the threads mechanisms, as described in section 5, that perfectly fulfil 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

[ADE94] B. Adelberg, H. Garcia-Molina, B. Kao, "Emulating Soft Real Time Scheduling Using Traditional Operating System Schedulers", Proceedings of the Real Time Systems Symposium, pp 292-298, San Juan, Puerto Rico, December 7-9, 1994

[AME94] P.D. Amer, C. Chassot, T. Connolly, P. Conrad, M. Diaz, "Partial order transport service for multimedia and other applications", IEEE/ACM transactions on Networking, October 1994, Vol. 2, No. 5

[BAK94] T.P. Baker, F. Mueller, V. Rustagi, "Experience with a Prototype of the POSIX Minimal Realtime System Profile", IEEE Newsletter on Real Time Systems, pp 29-33, 1994

[BER91] B. Berthomieu, M. Diaz, "Modelling and verification of time dependant systems using time Petri nets", IEEE transactions on software engineering, Vol. 17, No. 3, pp 259-273, March 1991

[BLA96] G. Blakowski, R. Steinmetz, "A media synchronization survey: reference model, specification, and case studies", IEEE journal on selected areas in communications, Vol.14, No 1, pp 5-35, January 1996

[CHA95] C. Chassot, M. Diaz, A. Lozes, "From the partial order concept to partial order multimedia connections", Journal of high speed network (JHSN), Vol. 5, No 2, pp 181-191, 1995

[COU94] G. Coulson, G. S. Blair, P. Robin, "Micro-kernel support for continuous media in distributed systems", Computer Networks and ISDN systems, Vol. 26, pp 1323-1341 (1994)

[DIA93] M. Diaz, P. Sénac, "Time stream Petri nets, a model for multimedia streams synchronization", proceedings of MultiMedia modeling'93, pp 257-273, Singapore, November 1993

[DIA94] M. Diaz, P. Sénac, "Time Stream Petri Nets, a model for timed multimedia information", 15th International Conference on Application and Theory of Petri Nets, Springer Verlag LNCS, R. Valette Editor, pp 219-238, Zaragoza, June 1994

[DIA95] M. Diaz, K. Drira, A. Lozes, C. Chassot, "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

[DIA97] M. Diaz, P. Owezarski, "From multimedia models to multimedia protocols", in Computer Networks and ISDN Systems, n° 29, pp 745-758, 1997

[JEF92] K. Jeffay, D.L. Stone, F.D. Smith, "Kernel support for live digital audio and video", Computer communications, Vol. 15, No. 6, pp 388-395, July / August 1992

[JEF94a] K. Jeffay, D.L. Stone, F.D. Smith, "Transport and display mechanisms for multimedia conferencing across packet-switched networks", Computer networks and ISDN systems 26, pp1281-1304 (1994)

[JEF94b] K. Jeffay, "On latency management in Time-Shared Operating Systems", IEEE Newsletter on real time systems, pp 34-38, 1994

[KAN94] 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

[KAT94] D.I. Katcher, K.A. Kettler, J.K. Strosnider, "Modeling DSP Operating Systems for Multimedia Applications", Proceedings of the Real Time Systems Symposium, San Juan, pp 287-291, Puerto Rico, December 7-9, 1994

[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

[OWE96a] P. Owezarski, M. Diaz, "Models for enforcing multimedia synchronisation in visioconference applications", proceedings of the 3rd MultiMedia Modeling conference - Towards the information superhighway (MMM'96), pp 85-100, World scientific editor, Toulouse, France, November 12-15, 1996

[OWE96b] P.Owezarski, "Conception et Formalisation d'une Application de Visioconférence Coopérative. Application et Extension pour la Téléformation", PhD Thesis dissertation, In french, University of Toulouse III, 154 pages, 1996

[OWE98] P. Owezarski, M. Diaz, C. Chassot, "A Time Efficient Architecture for Multimedia Applications", IEEE Journal on Selected Areas in Communication, Special Issue on Protocols and Architectures for Applications of the 21st Century, Vol. 16, N° 3, pp 383-396, April 1998

[SEN94] P. Sénac, M. Diaz, P. de Saqui-Sannes, "Toward a formal specification of multimedia synchronization", Annals of telecommunications, pp 297-314, May / June 1994

[SEN96] P.Sénac, M. Diaz, A. Léger, P. de Saqui-Sannes, "Modeling Logical and Temporal Synchronization in Hypermedia Systems", IEEE journal on selected areas in communications, Vol. 14, No. 1, pp 84-103, January 1996

[VOG95] A. Vogel, B. Kerhervé, G. Von Bochmann, J. Gecsei, "Distributed Multimedia and QoS : A Survey", IEEE Multimedia, pp 10-19, Summer 1995