Enforcing multipoint multimedia synchronisation in videoconferencing applications

Philippe Owezarski

LAAS-CNRS
7, Avenue du Colonel Roche
F-31077 Toulouse cedex 4
E-mail: owe@laas.fr

Abstract.

This paper deals with the problem of multimedia synchronisation in fully meshed multipoint videoconferencing applications. This work relies on previous work dealing with multimedia synchronisation in point to point videoconferencing: Up to now, the two main approaches are based on (1) timeline or on (2) temporal intervals composition. But, the first approach is not suited for multimedia data, and the second not suited for multipoint synchronisation. This paper then proposes a hybrid synchronisation solution taking advantage of the two preceding approaches. A multipoint videoconferencing tool based on this hybrid solution, called *Confort*, has been designed, developed, implemented and evaluated. This paper then deals with implementation details, as well as evaluation measurements.

1. Introduction

Videoconferencing is nowadays a mandatory tool for many real time collaborative working environments. A lot of videoconferencing tools already exist and among them it is possible to find industrial and commercial products as Netmeeting from Microsoft, or academic tools as VIC [MCC94] from LBL or Rendez-vous (the IVS successor) [TUR93] from INRIA. Such a situation is due to the fact that a large business related to desktop videoconferencing exists and software editors try to take as many parts as possible of this market. This market is increasing rapidly, pushed by strong needs in concurrent engineering (in industrial domains as aeronautic or automotive manufacturing, electronic design, etc.). But the first projects dealing with concurrent engineering, evaluated videoconferencing tools in actual experiments and showed that, nowadays, no collaborative working environment, and in particular, no videoconferencing application fulfil collaborative working requirements; such results have been observed in GDMD (Globally Distributed Microsystem Design) and CANET (Collaborative Automotive NETwork) [OWE97a].

In particular, the main problem with videoconferencing tools is their lack in term of QoS (Quality of service) guarantees. Among all the QoS parameters the most meaningful seems to be the audio quality that can currently range in a single work session from correct to incomprehensible (while it has to be constant), the end to end delay that has to be as short as possible to increase the interactivity level between

users, and all temporal synchronisation constraints. In fact, users are really disturbed when the temporal features of the audio and video streams are not enforced, as voice can be hard or impossible to understand, and the lack of lips synchronisation makes the audio / video correlation disappear. Besides, multimedia synchronisation is currently the key point to address and the most difficult to solve for designing multimedia systems [BLA96].

In related work, multimedia synchronisation is most of the time not addressed, and when it is, the selected solution is based on timestamps. This is the case for all videoconferencing applications using RTP [SCH96] as VIC, VAT or Rendez-Vous, even if it is now well known that the semantics of timestamps does not cover the requirements of multimedia synchronisation. Based on this observation, some work lead to another synchronisation approach based on temporal intervals composition. This approach has been selected and improved at LAAS, and lead to the design of a point to point videoconferencing tool: PNSVS (for Petri Net Synchronised Videoconferencing System). Therefore, this paper deals with the extension of PNSVS toward a multipoint videoconferencing application enforcing multipoint multimedia synchronisation. This new videoconferencing tool is called "*Confort*".

The remainder of this paper is as follows: first the problematics related to multimedia synchronisation is described (section 2). In particular, as it is already presented in the literature [BLA96], this section focuses more specifically on the problem of multiple users synchronisation. Then section 3 describes the principle of the two main synchronisation approaches that were spoken about just before. Given the analysis of these two approaches, and based on the experience gained from the PNSVS design and development, a hybrid synchronisation is presented (section 4). Using this hybrid approach, section 5 describes the implementation principles of *Confort*, and section 6 gives some experimental results and measurements. Finally section 7 concludes this paper.

2. Synchronisation problematics

2.1. Multimedia synchronisation

The problematics of multimedia synchronisation is closely related to the characteristics of the media to synchronise. In fact, in a videoconferencing tool, media (audio and video) are dynamic with strong temporal requirements. The problematics of multimedia synchronisation consists then in ensuring both the intra and inter streams temporal constraints. Ensuring the intra-stream temporal constraints consists in controlling the jitters on each audio or video object in order to keep it under acceptable maximum values. It means that the temporal profile of media objects has to be respected on the receiving site. For instance, a 25 images/s video stream has to be played back with a rate of 25 images/s. The only tolerance is called jitters: in fact, because of the non perfect human perception of audio and video, some temporal variations (jitters) can be accepted, a human user being unable to notice that one or two images have been lost on a 25 images/s video stream, or that an image presentation duration has been a bit longer or shorter than expected [STE96].

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 where lips movement has to be related to the voice stream).

2.2. Multipoint synchronisation

Multipoint synchronisation can be seen as multiple users synchronisation. This synchronisation task aims to avoid semantics incoherence in the dialogue between all users. In fact, delays can change significantly in distributed systems, and it is mandatory to avoid that one user receives the response to a question before the question itself. In collaborative systems, this kind of problem is generally solved using Lamport clocks that create in the system a logical order based on causal dependencies and that has to be respected by all users.

But, in a videoconferencing application that computes dynamic media, as audio and video, having strong temporal requirements, the Lamport clocks seem to be quite limited. In fact, in a videoconferencing tool, interactivity is of major importance, and the semantics is closely related to delay and explicit time. Generally, it is acknowledged that for a highly interactive dialogue, delays between users have to be less than 250 ms [JEF94]. Beyond this threshold, audio and video data can be considered too old and of no significance. Therefore multipoint synchronisation has to design and set up mechanisms allowing the control of end to end delays between users, in order to ensure the suited order between audio and video information, on a temporal basis.

3. Related approaches

As it has been mentioned in section 1, two main approaches exist in the literature. The first one, and the most used, relies on a timeline basis - this is an absolute time approach - while the second relies on the composition of time intervals - this is a relative time approach.

3.1. "*Timeline*" approach

This approach, consisting in using timestamps is easy and intuitive. In fact, each audio or video presentation object contains its own presentation date, and the presentation process has just to display it at the right time. This method ensure the intra and inter-streams synchronisation: a 25 images/s video sequence is replayed with a 25 images/s rate with respect of the temporal constraints on each object (intra-stream synchronisation), and the inter-streams synchronisation is also ensured, as each stream synchronises itself on a common temporal axe and are then synchronised the one related to the other.

However, timestamps do not solve the problem of acceptable jitters on information objects. In fact, timestamps are only a date on a time axe, and are then unable to

integrate the acceptable variations on the presentation of each object. This approach does not provide a suitable solution to the problem of multimedia synchronisation

3.2. Temporal intervals composition

The second approach frequently encountered in literature is based on temporal intervals composition [ALL83]. It means that the presentation time of one object depends on the preceding object presentation completion. Thus, only composing the time durations of all presentation data units can create multimedia synchronisation scenarios. This is the case for instance of the OCPN (Object Composition Petri Net) model [LIT90].

To design guaranteed synchronised multimedia, it is mandatory to consider all problems due to the temporal variability of the computing times (jitters, drift) and to link them to the properties inherent to each multimedia object (for instance to link the jitters of an operating system to the acceptable jitters 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 realised in this domain, and some models have been proposed. In particular, the OCPN (Object Composition Petri Net) model only considers nominal computing times, and does not address the temporal jitters acceptable on most of the multimedia data.

This limitation led [DIA93] [DIA94] to propose a new model, called TSPN for Time Streams Petri Nets, taking into account this temporal variability aspect. By definition, TSPNs use temporal intervals on the arcs leaving places. This allows authors to take into account 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 the computing 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.

Example: Using a TSPN for PNSVS

It is proposed in this part to use the TSPN model to describe the set of the timed synchronisation behaviours that appears in a videoconference application as PNSVS [BOY98].

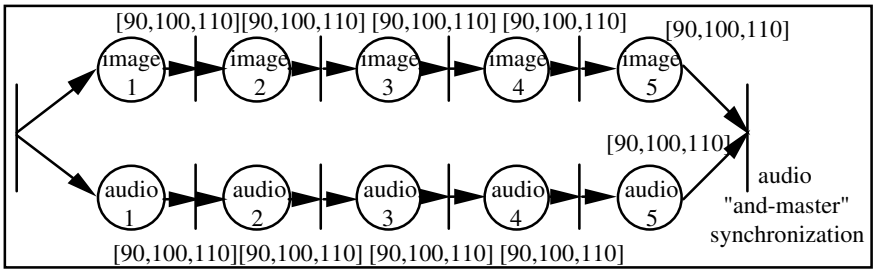

Fig. 1. Videoconference synchronisation constraints: TSPN based modelling

As an example, Fig. 1 describes the requirements for a videoconferencing system whose QoS parameters are:

- A throughput of 10 images per second. Then the nominal presentation time of a video object is 100 ms;
- An acceptable jitters on the audio or video of 10 ms (in absolute value) [JEF94]. It follows that the temporal validity intervals are [90, 100, 110] for one image and one audio packet;
- The audio is the most important media: inter-stream synchronisation is of «and-master» type, the voice being the master. In fact, voice being more important than video, it is defined as the master stream and its temporal constraints have to always be fulfilled; the «and» part of this 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 [JEF94]. 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. This is because the maximum drift on 5 objects is 50 ms and the inter-stream drift is then less than 100 ms.

As it has been shown before, as well as in [OWE98] for PNSVS, the temporal intervals composition approach is perfectly suited for point to point multimedia synchronisation. But this approach is not so powerful for multipoint synchronisation. In fact, it is quite difficult with such an approach to synchronise all users and to control the end to end delay as no global clock is used and we do not assume any temporal properties on the delays in the network. In fact, each jitter on an object introduces desynchronisation between users. As with Lamport Clocks, TSPN only allows logical synchronisation (or ordering) between users.

4. An hybrid solution for multipoint multimedia synchronisation

As on one side, timestamps are not suited for multimedia synchronisation, and time intervals composition not suited for multipoint synchronisation, it is not possible to

use one of these approaches. But on the other side, time intervals composition is perfectly suited to multimedia synchronisation, and timestamps are, until today, the best way to achieve multipoint synchronisation and a global time in distributed systems. It is then proposed in this paper to use an hybrid approach where:

- Multimedia synchronisation is achieved thanks to a time intervals composition mechanism;
- Multipoint synchronisation is achieved thanks to the set up of a global clock in the distributed system.

Concerning multimedia synchronisation in *Confort*, the principle is very similar to the one of PNSVS described in [BOY98], as TSPN and more generally synchronisation mechanisms relying on temporal intervals composition are designed for 1 to N synchronisation schemas. Moreover, this approach has been designed for unpredictable distributed systems where delays, loss level, etc. are changing. In fact, the synchronisation constraints to enforce on each object depend on the presentation duration of the media object, and on the arrival date of the considered object. The synchronisation scenario modelled by the TSPN of Fig. 1 is refined, concerning actual presentation dates, each time a new media object reaches the receiving videoconference entity: this is an a posteriori approach for synchronisation. In addition, as current general operating systems are asynchronous (as Unix, Windows, etc.), performing temporal synchronisation on media objects in low level layers is useless, as synchronisation features are disturbed when media objects cross communication and operating system layers. The synchronisation task has then to be put at the upper level of the receiving entity, in the application layer (according to OSI or Internet models). As the synchronisation task is put on the receiving entity, synchronisation approaches based on temporal intervals are then ideally designed for 1 to N multimedia synchronisation schemas.

Given the TSPN of Fig. 1 (or any other), the synchronisation task on each receiver has only to implement a synchronisation runtime able to enforce the synchronisation constraints modelled by the TSPN, and to schedule media objects adequately. The principle of such a runtime is described in section 5, which deals with implementation issues.

Concerning multipoint synchronisation and given a global synchronised clock over all the distributed system, it is then very easy to control end to end delay and causality in media object deliveries. In fact, it is sufficient to put in the header of media objects a timestamp (related to the global clock) and to control that the time between the sending and arrival dates is not greater than the maximum end to end delay required. This approach is very similar to the one of RTP (Real time Transport Protocol) that is recommended by IETF for audio / video Transport [AVT00], except that we here assume a global clock. How distributed workstations clocks are synchronised to obtain this global clock is explained in section 5 (related to implementation issues).

Finally, with the temporal intervals composition based approach that allows 1 to N multimedia synchronisation schemas and the synchronisation of all the workstation clocks - and therefore of all the sending entities of the videoconferencing session - an N to N synchronisation schema is achieved.

5. Implementation issues

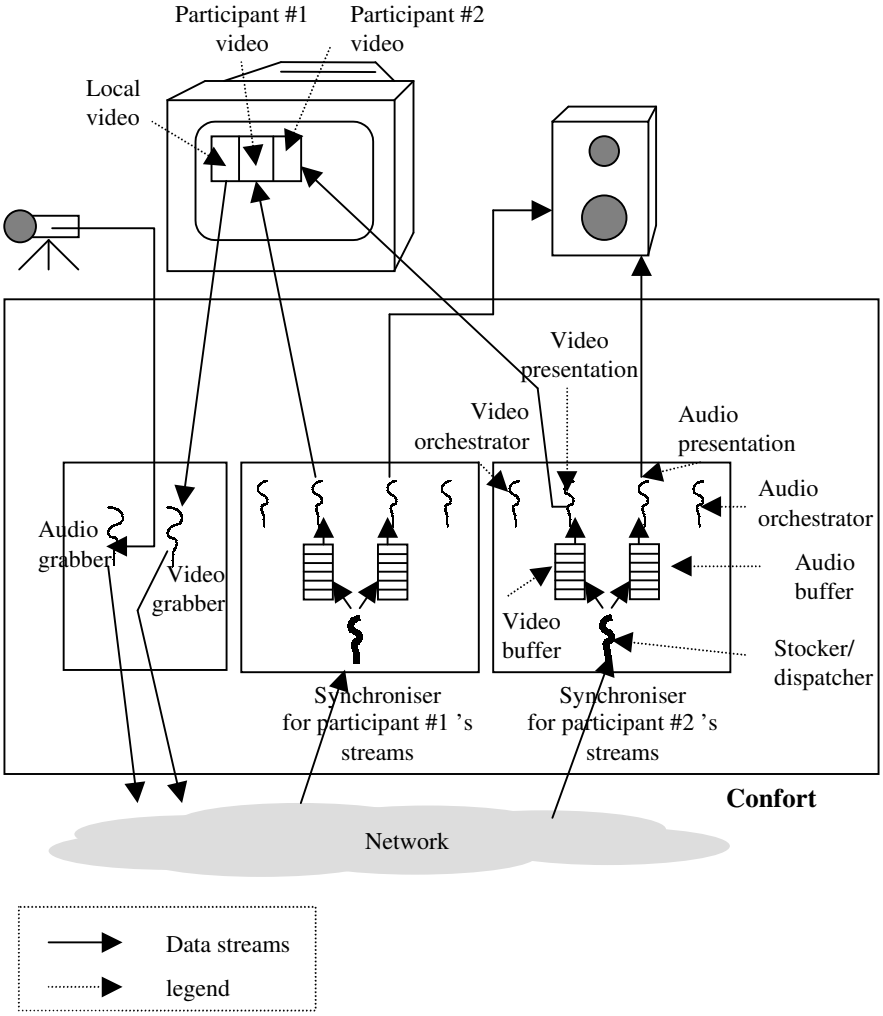

Fig. 2: Confort architecture (3 participants)

This section deals with implementation issues of the hybrid synchronisation task described in the preceding section. Thus, it is shown:

- How multimedia synchronisation can be enforced with a strong respect of audio and video temporal constraints?
- How the global clock can be set-up on a potentially wide area distributed system?

- And how some other functionalities as openness and auto-configurability can be integrated in Confort, in order to make it rapidly and easily usable in any collaborative working environment?

To illustrate all these implementation issues, Fig. 2 depicts the *Confort* software architecture.

5.1. Multimedia synchronisation implementation principle

An important constraint of the Confort development concerns its ability to work on general systems as Windows or Unix, with general hardware. We prefer to avoid dedicated systems and hardware architectures that are not widely available, a videoconferencing tool having to work on as many desktop computers as possible.

However, implementing multimedia synchronisation mechanisms having to enforce temporal constraints requires some real time functionalities of the operating system, to be able to control the processes scheduling. For this reason, Solaris has been selected as it provides a real-time scheduling class, and it was the only one to provide this feature in 1994, when the work on videoconferencing began at LAAS. This real time scheduling class (also called RT) is really interesting as it gives RT priorities to user processes greater than the ones of system tasks, and also a predictable and configurable scheduler [KLE96].

Before addressing the multipoint synchronisation problem, some work has been done, at LAAS, to enforce synchronisation in the PNSVS point to point videoconferencing application [OWE96]. And it has been said in section 4 and in [OWE97b] that synchronisation mechanisms are located in the top layer of the receiving machine. Thus, the multimedia synchronisation problem can be solved in Confort as in PNSVS. In fact on each 1 to N multiconnection from one sender to all other participants of the videoconference session (the receivers), we can duplicate the synchronisation automaton of PNSVS on each receiving entity.

Finally, as it is depicted on Fig. 2, each videoconferencing entity (on each machine) consists of:

- A sending task that grabs video and audio objects and multicasts them to all the other videoconference participants;
- For each distant participant, a synchronisation task that synchronises the audio and video streams coming from this distant participant. Then with N participants, each entity will run N-1 synchronisation tasks similar to the one of PNSVS.

As a reminder, the following recalls how the PNSVS intra and inter-streams synchronisation mechanisms are working. In fact, each stream is computed by a dedicated process (implemented thanks to a thread). But before being presented (on the screen for images or loudspeakers for audio), each object has to be computed by a specific hardware (video or audio board) managed by the kernel and with system scheduling class and priority. And there is no guarantee on the time that the computing of each image and audio object will take: this is a typical asynchronous system. It is then required to create a second process (thread), in the RT class, whose function is to control the temporal behaviour of the presentation process. Running

with the RT priority, the control process, also called orchestration process, can perfectly control the associated presentation process. It can stop (or kill) it if it overruns the maximum presentation duration, or it can also make it sleep until the minimal presentation time is reached: the intra-stream synchronisation constraints are therefore enforced. Then, as depicted on Fig. 2, each synchronisation task for synchronising audio and video streams consists of 4 threads:

- One audio presentation thread;
- One video presentation thread;
- One audio orchestration thread devoted to the control of the temporal behaviour of the audio presentation thread;
- One video orchestration thread devoted to the control of the temporal behaviour of the video presentation process.

On the other side, inter-streams synchronisation is performed between the orchestration threads, that have to synchronise themselves on a rendezvous point (corresponding to inter-streams transition on the TSPN model, as the one of Fig. 1 for instance), and this with respect to the semantic rule of the inter-streams synchronisation. For example, if the inter-streams synchronisation rule is "and-master" (Fig. 1), the synchronisation constraints on the "audio 5" object must be guaranteed. Then, if the video presentation thread returns first, the video orchestrator will then wait the audio threads. When these laters return, the transition is fired and a new synchronisation period begins (with an inter-streams drift of 0). If the video presentation thread is late, then, the audio orchestrator waits it until its maximum presentation time. The transition firing is then caused by the video threads completion or the time-out of the audio timer. In that case, the audio orchestrator forces the video presentation process to stop its current task, and to begin the one of the first image of the following synchronisation period.

5.2. Clocks synchronisation

Concerning clocks synchronisation, my first work consisted in investigating for existing solutions. Many are related in the literature, and they are always based on temporal information exchanges, over communication networks. But it also appears that none of these solutions do provide any guaranty on the precision they can reach on a general network as the Internet. Even NTP3 [MIL92], that has been designed for synchronising clocks over the Internet, does not provide any guarantee, the synchronisation precision being dependant on the network load and RTT, for example. Nevertheless, on LANs, NTP proved to be really efficient, as LAN provides large bandwidth and low RTT. NTP is then a suited solution for synchronising clocks on a LAN and has therefore been selected. This choice is one among a lot of similar solutions that are as efficient as NTP. The advantage of NTP, compared to other solutions is its availability: many free implementations are available, and this protocol is already widely deployed.

Nevertheless, it remains to solve the problem of synchronising machines on WAN. Up to my knowledge, there is no clock synchronisation protocol really efficient on WAN as the Internet. The selected solution is then the one offered by GPS, which transmit the universal time of reference atomic clocks by satellite links, with a precision of few microseconds. Thus, the solution retained consists in setting one GPS

board on each LAN, and then to synchronise all the clocks of all the machines connected to this LAN, with the clock of the GPS board, using NTP. Similar solutions are sometime used for scalable and reliable multicast protocols. Thanks to this approach it is possible to synchronise very precisely the clocks of partners involved in a widely distributed videoconferencing session. Section 6 presents some evaluation measurements of this solution.

5.3. Additional functionalities of Confort

The other functionalities that have to be integrated in *Confort* are not related to synchronisation. The problem here concerns the ability of the videoconferencing tool to be easily integrated in a more general and generic tele-working environment. These kinds of environments may or may not manage the work session membership, and may or may not send control and/or command messages. Depending on the work session control tool of the collaborative environment (maybe a workflow management system), the videoconferencing system has to:

- Be able to receive control and/or command messages from the work session controller, and then to dynamically change the group of participants (by adding or removing some members), change the requested QoS on audio and video streams, etc. This ability is called "openness". For that an interface has been designed, that defines the type of control/command messages, and fields they contain;
- Be able to detect itself membership changes. This ability is called "auto-configurability". Then, if the videoconferencing receiving entity detects that new audio and video streams arrive from a new participant (that were not participating in the session up to now), it automatically allocates the required resources (buffers), starts the synchronisation entity (consisting of 4 threads), and create a video window to display the received images. On the other side, if the receiving entity does not receive any data from one participant for a long time, this participant is removed from the active members, the allocated resources are freed, and the associated synchronisation threads are killed. This principle allows the application to always have a coherent knowledge of the state of participants group membership, even in case of failures or crashes.

The architecture principle to develop such functionalities relies on the use of a stocker/dispatcher thread (Fig. 2). This thread is the master of the overall architecture as, depending on the type of data it receives, it can:

- Store audio and video data in the right buffer, depending on who send it;
- Change what has to be changed if it receives a control/command message (openness);
- Adapt itself the dynamic architecture of the videoconferencing entity if it notices that a new participant is sending data, or at the opposite, if a participant does no more send data (auto-configurability).

6. Results

Confort has been implemented on Sun workstations (Sun Sparc Station or Ultra station), in C++, using the Solaris 2.5.1 or 2.7 operating system. Confort supports Parallax or SunVideo video boards that allow the system to grab, display, compress and uncompress images using the M-JPEG compression algorithm. Audio is the Sun standard audio system. Workstations running Confort communicate using the UDP transport protocol. Tests have been performed on top of a 10 Mbps Ethernet and a 155 Mbps ATM network.

Confort is a full meshed and full duplex videoconferencing application that can process up to 25 images/s (320 x 240 pixels and 24 bits coded colours).

This part aims to evaluate the performances of the hybrid multipoint synchronisation task developed in *Confort*. First it is required to verify if all synchronisation solutions do respect the temporal presentation requirements, and in particular the quality required for the intra and inter-stream synchronisation. The measurements reported here have been realised in the case of a 10 images/s videoconference application whose synchronisation requirements were modelled by the presentation TSPN given in Fig.1, and with a 3 users videoconferencing session.

Fig. 3. Shows, for the first 100 images received by one of the user from another, the jitters that appear during the presentation of each image. The measured jitters are the difference between the effective presentation duration and the nominal presentation time. This Fig. clearly shows that the maximum jitters value of 10 ms is always fulfilled. Note that the jitters is always negative: this is because in this experiment there is no network problems (no loss and no jitters, the goal being to evaluate the temporal synchronisation mechanisms) and then the needed data are available as soon as the presentation process needs them. Thus, the anticipation mechanism of the temporal intervals composition approach, that stops the presentation of an object as soon as its minimal presentation time has been reached, actually works. The very small variations are due to the real-time scheduler of Solaris 2.

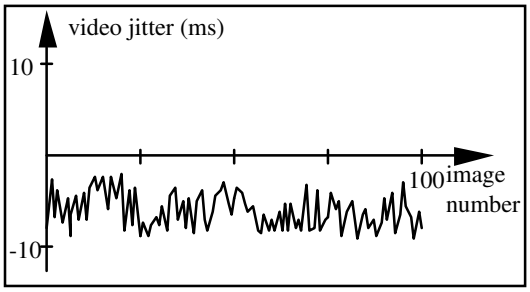

Fig. 3: Measurement of the jitters on *Confort* images

Fig. 4 shows the same measurements but now for the Confort audio stream. As in the video case, intra-stream synchronisation requirements are always fulfilled. Note that now the jitters are always positive. In fact, this is because firing the intra-stream transition is controlled by the audio driver signal. With a time scale expressed in millisecond, the presentation duration of one audio object is always equal to 100 ms; variations are due to the time required by the system to take into account this information.

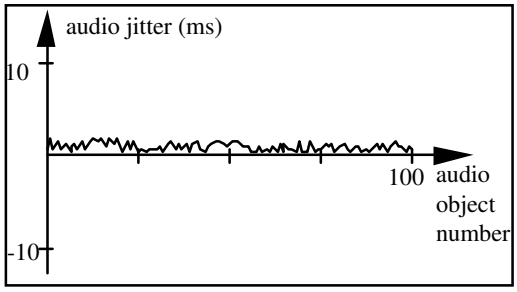

Fig. 4: Measurement of the jitters on *Confort* audio objects

Finally, Fig. 5 shows the curve representing the inter-stream drift for the audio and video sequences. Fig. 3 shows that the video jitters is always negative and Fig. 4 that the audio jitters is always positive. Thus, the inter-stream drift (the difference between the audio and video presentation dates) is positive. It can be easily seen that this drift increases during each period of 5 objects (what is due to the cumulative effect of the jitters) and it is then reduced to a value near zero at each inter-stream synchronisation. It clearly follows that the inter-stream synchronisation requirements are perfectly fulfilled: the maximum drift never exceeds 50 ms, the maximum allowed value being 100 ms. Note that the maximum value is never reached here because the audio jitters is always very close to zero and thus the inter-stream drift is essentially due to the video jitters.

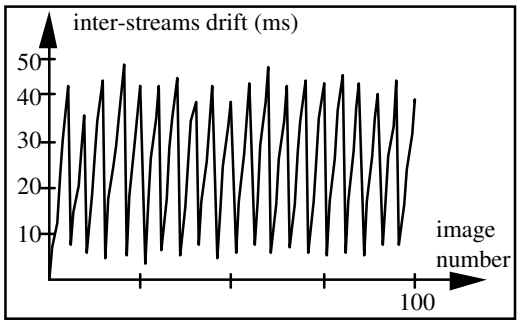

Fig. 5: Measurement of *Confort* inter-streams drift

Therefore, the evaluation of multimedia synchronisation mechanisms proves that they perfectly work. It remains, nevertheless, to evaluate the mechanisms allowing the synchronisation of the clocks of all the workstations involved in the videoconferencing system. These measurements have been performed on a dedicated 155 Mbps ATM local network, on which delays are perfectly known (at a 1 ms scale), and thanks to a network analyser. It appears that the desynchronisation between the three workstations clocks is always less than 2 ms, and this drift between clocks is essentially due to temporal variations in the transport layer, supported by a non real time operating system, and is not due to low level network layers. NTP, then, provides a suited solution for synchronising workstation clocks on a LAN. In fact, at the scale of a videoconferencing application, such a drift between clocks is not significant, and human users cannot notice an end to end delay improvement of 2 ms.

Nevertheless, *Confort* has not been tested on general wide area network as the Internet (it has only been tested on a national private network: SAFIR), and then we do not perform any measurement in such case. But, GPS manufacturers ensure that the precision of synchronisation with GPS boards (connected to reference atomic clocks by GEO satellite links) is of few microseconds. It is then of no significance compared to the precision of NTP on a LAN, and compared to the requirement of a videoconferencing application.

7. Conclusion

This paper has presented a new hybrid approach for synchronising multimedia streams in a multipoint videoconferencing application. The multimedia synchronisation relies on a temporal intervals composition basis to take into account the jitters that are acceptable on audio and video objects. Then, the interactivity and the coherence of multimedia information between distributed participants is achieved thanks to a clocks synchronisation mechanism. Taking advantage of a global distributed clock, it is then possible to perfectly control the end to end delay and its variations, and to enforce mulipoint synchronisation. The solution for synchronising distributed clocks is very practical as it relies on dedicated hardware (GPS) and NTP. Nevertheless, it has been proved that the hybrid solution proposed in this paper is really efficient.

The limitation that appears is related to the portability of the Confort videoconferencing tool and the management of its evolutions. In fact, this tool is written in C++ because it was required to access the real time functionalities of Solaris. But, each time the video board or the video drivers version change, it is required to reopen the source code, and to modify the video procedures. To avoid such portability and maintenance problems, it would be interesting to use generic video interfaces as the Java JMF, for instance, that solves all the compatibility and portability problems of video programs. But, today, the mapping between the Java Virtual Machine (JVM) and the real time services of the operating system is not done, the JVM having a really asynchronous behaviour. Nevertheless, the Java approach seems really interesting for portability and version maintenance. That is why next step

will consist in studying the mapping between the Java runtime and the operating system kernel services.

8. References

[ALL83] J. F. Allen, "Maintaining knowledge about temporal intervals", Communications of the ACM, Vol. 26(11), November 1983

[AVT00] Audio Video Transport Group of the IETF, available at http://www.ietf.org/html-charters/avt-charter.html, March 2000

[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, January 1996

[BOY98] M. Boyer, P. Owezarski, M. Diaz, " Dynamic QoS renegociation in the PNSVS videoconferencing application", International conference on Interactive Distributed Multimedia Systems (IDMS'98), Oslo, Norway, September 8 - 11th, 1998

[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

[JEF94] 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 (1994)

[KLE96] S. Kleiman, D. Shah, B. Smaalders, "Programming with threads", 534 pages, Prentice Hall 1996

[LIT90a] 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

[LIT90b] T.D.C. Little, A. Ghafoor, "Network Considerations for Dist'd Multimedia Objects Composition and Communication", IEEE Network Magazine, November 1990

[MCC94] S. McCanne, V. Jacobson, "VIC - Video Conference", Unix manual pages, November 1994

[MIL92] D.L. Mills, "Network time protocol (version 3): Specification, implementation and Analysis", RFC 1305, March 1992

[OWE96] 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

[OWE97a] P. Owezarski, D. Wartelle, P. Perrot, G. Ségarra, S. Guillouet, K. Drira, A. Meftah, M. Diaz, "Assessment methodology of new technologies for collaborative automotive design", 2nd International distributed conference on network interoperability, Madeira, Portugal, June 16-18th, 1997

[OWE98] P. Owezarski, M. Diaz, C. Chassot, "A Time Efficient Architecture for Multimedia Applications", IEEE Journal on Selected Areas in Communications, special issue on Protocols Architectures for 21st Century Applications, vol. 16, No. 3, April 1998

[OWE97b] M. Diaz, P. Owezarski, "From multimedia models to multimedia protocols", in Computer Networks and ISDN Systems, No 29, pp 745-758, 1997

[SCH96] H. Schulzrinne, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996

[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

[STE96] R. Steinmetz, "User Perception of Jitters and Media Synchronization", IEEE Journal on Selected Areas in Communications, vol. 14, number 1, pp. 61-72, January 1996

[TUR93] T. Turletti, "H.261 software codec for videoconferencing over the Internet", Research report INRIA No. 1834, January 1993