Synchronization loss and concealment mechanisms in distributed multimedia application[1]

R.Castanet, B. Cousin, C.Viho
IRISA
Campus de Beaulieu
35042 Rennes cédex
FRANCE
email : bcousin@irisa.fr

Abstract

One of the fundamental problems in the domain of distributed multimedia applications is the synchronization of the communication elements. Multimedia data is defined as *bundles* composed by several *streams* (video, audio, or text streams). A stream can be defined as a sequence of informations units (IU), for example : image frame, line of text, sound unit, etc). Intra-synchronization binds IUs of the same stream with ordered or temporal constraints (eg one image each 1/25th second), when inter-synchronisation binds streams of the same bundle (eg lip synchronization between image and voice). The UIs of the streams will be transmit by the transfert level.

In the network environment considered in our work, three specific phenomena may produce synchronization loss : load variation, jitter (delay variation), packet loss. It is necessary to introduce mechanisms upon the Transfert layer to preserve and/or restore the synchronization of the transmitted streams (according to application requirements). Based on single data stream, usual mechanisms, used to preserve synchronization, do not take in account the particularities of multimedia streams. Generally, usual solution drops data arriving during the non synchronization phases (between the moment where the loss of the synchronization is detected and the moment where the synchronization is resumed).

In this paper, after the presentation of the synchronization services in distributed multimedia application architecture, we propose formal definitions of inter-stream and intra-stream synchronization. Then we present the influence of the loss of the intra-stream synchronization (inside each stream) and inter-stream (inside a bundle of streams). We show how to use the specification of the synchronization constraints (inter and intra) to propose some mechanisms, that are named concealment mechanisms. Concealment actions provide services that conceal the erroneous behavior of the original synchronization service. After synchronization loss, depending on the type of constraints of the multimedia bundle and streams, corresponding actions are given to achieve synchronisation recovering.

1. Introduction

Hardware for processing digitized audio and video and high speed networking are becoming available at low cost. These developments have stimulated interest in teleconferencing applications based on desktop personal computers. In this paper, we focus on the problem of support for communication of the audio and video portions of a computer-based conference. There are a large number of research and commercial product development activities that seek to provide solutions. Many of these involve application of specialized technology develop for video teleconferencing and standardized in the CCITT H320 series. When these systems rely on ISDN with dedicated communication link, in contrast our research is concerned with solutions using networks based on asynchronous communications (Ethernet, FDDI, or ATM).

One of the fundamental problems in the domain of the distributed multimedia communication is the synchronization of the communication elements. Synchronization can be handled at different levels : cooperation, synchronization, and transfert levels. At the

---

[1] This research has been supported (under grant 92-1B-178) as part of CESAME project, a CNET-CNRS collaborative project on High Speed Network and Distributed Multimedia Systems.

cooperation level many roles can be defined : organizer (initiating the communication, managing the resources and the participants, etc), manager (controlling the communication, opening and closing the communication, authorizing late admissions, giving rights and duration for transmission, etc), sender (producing and sending information), receiver (receiving, storing and delivering information). A *session* is composed of all the participants, the synchronization constraints between the participants, all the resources needed for the communications, the media and the constraints between the senders and the receivers. A session is handled into the cooperation level.

Synchronization can be handled at different levels : cooperation, synchronization, and transfert levels. At the cooperation level many roles can be defined : organizer (initiating the communication, managing the resources and the participants, etc), manager (controlling the communication, opening and closing the communication, authorizing late admissions, giving rights and duration for transmission, etc), sender (producing and sending information), receiver (receiving, storing and delivering information). A *session* is composed of all the participants, the synchronization constraints between the participants, all the resources needed for the communications, the media and the constraints between the senders and the receivers. A session is handled into the cooperation level.

Multimedia data to be transmitted can be seen as a *bundle* composed by several *streams* (video, audio, text, etc streams). A stream can be defined as a set of ordered and/or timed sequence of informations units (IU). For example : image frame, line of text, sound unit, etc)[CCS93]. A bundle will be handled by the synchronization level of the figure 1, and the streams by the transfert level.

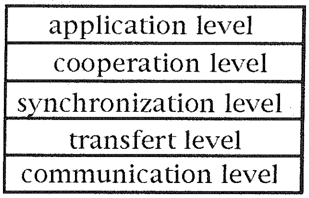

Figure 1 : architecture of distributed multimedia application

Three remarks can be done about the figure 1 [DIA92], representing a possible organization by levels for a distributed multimedia application :

- Each level deals with only one main object (session : cooperation level, bundle : synchronization level, stream : transfert level), the communication level uses some high speed network (for example ATM [PRY93])
- Figure 1 presents a stack, but for some multimedia communication, all the stack is not useful, the services offered by the lower layers of the stack can be directly used, depending of the needed services.
- Each level deals with synchronization constraints : session : between the participants, synchronization : between the streams of a bundle, transfert : between the IUs of a stream.

Some Transfert mechanisms are used for the preservation of the stream synchronization. However, it can appear that, in spite of the implemented mechanisms, synchronization loss is observed due for example to the delay jitter, or to unexpected stream interruption, or to the congestion on the transmission network. Generally, the chosen solution is ignoring the data that arrive during the non synchronization phases (between the moment where the loss of the synchronization is detected and the moment where the resumption is handled).

In this paper we present the influence of the loss of the intra-stream synchronization (inside each stream) and inter-stream (inside a bundle of streams). We show how to use the specification of the synchronization constraints (inter and intra-streams) to propose some mechanisms, that are named concealment mechanisms, dealing to conceal at best the non synchronization phases.

2. Presentation of the study

A stream can be seen as a set of information units bound by synchronization constraints [CCS93][STE90]. We suppose that a transmission service is available for the stream transfer between several entities on distant places. However, the transmission of a multimedia data flow through an asynchronous network generates temporal "shift" between the information units. So, it is necessary to use, above the transfert service (and below the application level) mechanisms to preserve (according to the application entities) the synchronization constraints of the transmitted flow [RRV90][SDF93]. In a multimedia architecture, the role of the communication system is to assure the transfer of a bundle (with several streams) proposed by an application entity and to correctly present to the entities of the application level a bundle coming from the transport level (figure 3). In terms of service, we can say "how to respect the synchronization constraints of the streams for the delivered bundle ?".

It is necessary to introduce mechanisms upon the Transport service to restore (according to application requirements) the synchronization of the transmitted streams. The figure 2 presents the position and the tasks of the synchronization level to ensure the transmission of a submitted bundle and to deliver correctly (according to the synchronization constraints) the received bundle.

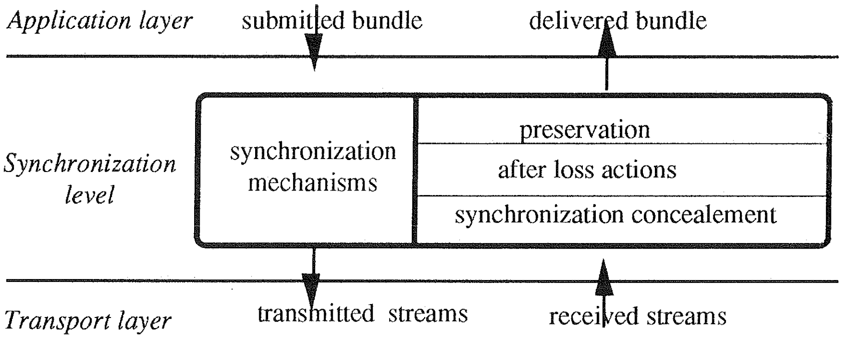

Figure 2 : actions of the synchronization level

Many mechanisms [RRV90][SDF93][STE90][ANH92] have been defined to maintain the synchronization between streams. Due to severe phenomena like jitter, unexpected interruptions of the flow, congestion on the transmission network during the data transmission basis, synchronization loss can happen . The loss of intra stream synchronization can be defined as the instants where a non respect of the synchronization constraints is observed between the IU of the streams, and in spite of the mechanisms build for the maintain of the synchronization.

The loss of the inter stream synchronization is the instant where a non respect of the synchronization constraints on the streams of a bundle is observed.

The communication system has to propose mechanisms to recover the synchronization ant to conceal the non synchronization in the interval between the loss and the recovery (figure 3).

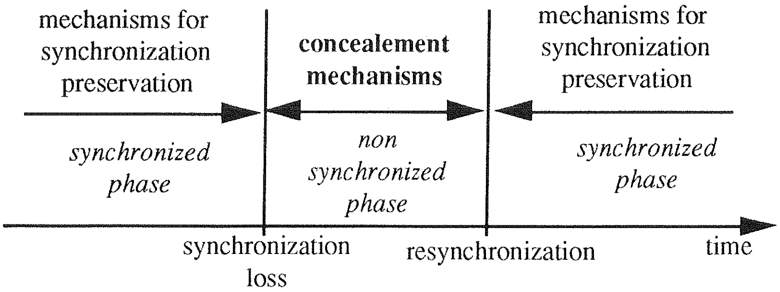

Figure 3 : synchronization mechanisms

In this study, we deal only with point to point communications. We try to present formally the loss of synchronization (inter and intra stream). For each type of loss, we propose concealment mechanisms and their formal definitions. At the end, we show how to use the synchronization constraints on the streams to realize the synchronization recovery.

3. Intra stream synchronization services and characterization of synchronization loss

Synchronization constraints on IU of a stream can be defined with preserving a certain order and/or a certain temporal signature. For example, if we consider a complete image of a video stream as an IU, then temporal signature can be expressed by : " one image per 1/25 th second", and preservation of order can be expressed by : " images are to be presented one after the other, according to the order of the sending". Combining these two expressions defines the synchronization constraint for a video stream .

The synchronization level depends on the definition level of the IU, but this is not our aim here. Different definition levels can be proposed for an IU (text stream : one character, word, sentence, etc; image stream : one pixel, line, full image (compressed or not), etc: sound stream : elementary sound, syllabi, word, piece of a musical melody, etc. In [STE90], different definition levels of an IU are given by a table. We suppose that a description level is chosen for the IU definition.

We note A_k, the kth IU of a stream. The stream A is the suite builds with the set of its IU, noted : $A = <A_k>$.

3.1 Expression of the intra-stream synchronization constraints

A constraint stream is a stream with a constrained synchronization. A synchronization constraint C on a stream A can be expressed with a constraint on each information unit A_k of the stream :

$$C(A) = <C(A_k)> \qquad (1)$$

In the above expression $C(A_k)$ can be the absolute presentation date that is suitable for the IU A_k, or another expression (for example with temporal dependence) describing the constraint that has to be applied on A_k in order to respect the synchronization constraint C on the global stream A . At first, we suppose that $C(A_k)$ is an absolute date. For example, the preservation of the temporal signature on a stream A can be expressed by the date when the IU have to be delivered. Then, $C(A_k)$ represents this date for the IU A_k of the stream A.

If we note $P(A_k)$ the effective presentation date of the IU A_k, then the preservation of the temporal signature on the stream A is :

$$\forall k, P(A_k) = C(A_k) \qquad (2)$$

We note $R(A_k)$ the effective reception date of the IU A_k. To simplify our purpose, we suppose that the value of presentation duration is constant and equal to δ ($\forall k, P(A_k) = R(A_k) + \delta$). Here, we assume appropriate services, especially those for real-time scheduling of tasks, are avaible in the underlying operating system.

This condition (2) will be always satisfied if we suppose that the transmission duration is constant (i.e. the network is able to provide a synchronous service). This assumption is not realist because the transmission duration in usual networks depends on the load of the network. Consequently, a synchronization loss is observed.

3.2 Characterization of the intra-stream synchronization loss

For an IU A_k of a stream A, we note $r(A_k)$ as the predicate that is true if A_k is received ($\bar{r}(A_k)$ if A_k is non received). We remark that if r(Ak) is true, a value of the effective reception date $R(A_k)$ is available.

An intra-stream synchronization loss (loss-S(A_k)) at the date t on an IU A_k is observed if and only if

$$\text{loss-S}(A_k,t) \Leftrightarrow (\bar{r}(A_k) \text{ and } (t > C(A_k)-\delta \text{ or } (r(A_k) \text{ and } R(A_k) > C(A_k)-\delta) \qquad (3)$$

The above definition implies the instantaneous detection of intra-stream synchronization loss. The first part of the definition explains that A_k is not received and the presentation date

$C(A_k)$ wished for A_k is out of date. The second part of the definition explains that the A_k is received, but later than the wished presentation date.

The a posteriori detection (when the stream is over) doesn't take in account the instant of the detection and can be expressed by :

$$\text{loss-S}(A_k) \Leftrightarrow (\bar{r}(A_k) \text{ or } (r(A_k) \text{ and } R(A_k) > C(A_k)-\delta) \quad (4)$$

That means that A_k has never been received or that the reception was too late for the wished presentation.

We have to consider also the order presentation (for example the projection of a set of slides needs the respect of the order of the slides). The order preservation implies that an IU A_k has to be presented before A_{k+1} for any k :

$$\forall i, j \,/\, i < j \Leftrightarrow P(A_i) < P(A_j) \quad (5)$$

When the synchronization loss is observed, mechanisms for the synchronization recovery have to be started. The usual behavior is to do not take in account the IUs that arrive in the interval between the loss and recovery dates. However, it is possible to treat in parallel this non synchronization phase and the recovery (in taking in account the synchronization constraints). The objective is to conceal the non synchronization by providing a synchronization service that is the nearest than the initial global service for the communication. This is why we call these mechanisms : concealment mechanisms.

3.3 Intra-stream non synchronization phases and concealment mechanisms

We define the presentation function X that associates a given date to the IU of a stream presented at this date. A presentation function that preserves the constraint C on the stream A verifies the relation : $\forall k$, $X(C(A_k)) = A_k$. If we suppose that the constraint preservation is true, then $P(A_k) = C(A_k)$. Given the hypothesis that the presentation duration is constant, $P(A_k) = R(A_k)$. Then, at the instant $t = C(A_k)$, $r(A_k)$ is always true. So, $X(t) = X(C(A_k)) = A_k$.

The presentation function is specified by :

$$X(t) =: \text{if } t == C(A_k) \text{ then if } r(A_k) \text{ then } A_k. \quad (6)$$

When concealment mechanisms and synchronization recovery are not available the function X describes the maintain mechanisms of the synchronization. The expression (6) means that only the IU received at time are delivered. It representes the best situation when the IU are received at time. So, if a synchronization loss is observed at the instant t, no defintion is associated to X(t).

In this case it is necessary to to express the synchronization loss in the presentation function. The concealment mechanisms associate a value (an IU) to $X(C(A_k))$ when A_k is not received at time.

The presentation function is specified by :

$$X(t) =: \text{if } t == C(A_k) \text{ then if } r(A_k) \text{ then } A_k \text{ else concealment\_action}(t). \quad (7)$$

Different concealment mechanisms can be proposed : delivering the last IU correctly received, delivering an interpolation of the last IU, etc.

Let us consider the concealment mechanism that delivers the previous IU. In this case the presentation function is :

$$X(t) =: \text{if } t == C(A_k) \text{ then if } r(A_k) \text{ then } A_k \text{ else } A_{k-1}. \quad (7\text{bis})$$

Unfortunately, this concealment mechanism is too weak and not very efficient.We are not sure that the previous IU has been correctly received. We have to refine this mechanism to catch the last correctly received IU.To do that, a function "last-safe" gives the last IU A_j that has been received without any synchronization error.

$$\text{last-safe}(A,t) =: A_j \,/\, \overline{loss}\_\bar{S}(A_j,t) \text{ and } ((\forall\ i>j \,/\, (C(A_i) < t, \text{loss\_S}(A_i, C(A_i)). \quad (8)$$

A presentation function that uses the above function is defined as :

$$X(t) =: \text{if } t == C(A_k) \text{ then if } r(A_k) \text{ then } A_k \text{ else last\_safe}(A,t). \quad (9)$$

This new concealment mechanism depends on the type of the coding used for the stream: I, B or P.

(1) Independant coding (as in M-JPEG[JPEG]).

In this case the coding/decoding of each IU are independant. We denote the type of the IU as I. The stream contains IU of type I. If a synchronization loss is observed on an IU, the mechanism (8) gives the previously deliver IU. For example with a sound stream, the effect of (8) gives a continuity feeling. On contrary, when not any concealment mechanism is available a jerky feeling appears.

(2) Differential unidirectional coding(as in H261[H261]).

In this case the coding/decoding of an IU may depend on the previous IU. We denote this type of information P.If a synchronization loss is observed on an IU, the previous correctly delivered IU (type I or P) is delivered. The synchronization recovery can be achieved only on an IU of type I.

(3) Differential bidirectional coding (as in MPEG[MPEG2]).

In this case the coding/decoding of an IU may depend on the previous or next IU. We note this type of information B. If a synchronization loss is observed on information I or P, the previous method is applied. If a synchronization loss is observed on information B, the concealment mechanism (8) gives an information I or P. Therefore, an IU of the type B depends on a previous or next IU and a synchronization loss can be appear only when a non reception of a future (not at the right time) is detected. So, the synchronization loss appears on the first IU of the type B (after an IU I or P). The algorithm (8) leads to deliver the previous information I or P. In this case, the disadvantage is that the delivered IU can be "old". This is the case in the MPEG-video [MPEG] when the interval between two IU of type I or P is too large (in this case it is possible to fix the number of the units of type B).

4. Inter-stream synchronization services and inter-stream concealment mechanisms

A set of multimedia informations can be seen as a bundle with different streams (image stream, sound stream, text stream, etc). The synchronization service has to achieve the synchronization between all the streams according to the application requirements.

4.1 Bundle definition, coupling notion and slice notion

A coupling express the synchronization constraints between the IU of different streams representing together a bundle. All the IU of several streams associated with a coupling represent a bundle slice. So, a bundle is defined as the sequence of its slices.

As the IU represents the elementary object of a stream, a slice represents the elementary object of a bundle.

We note F_k the k^{th} slice of a bundle F. A bundle F can be defined as :

$$F = \langle F_k \rangle \qquad (10)$$

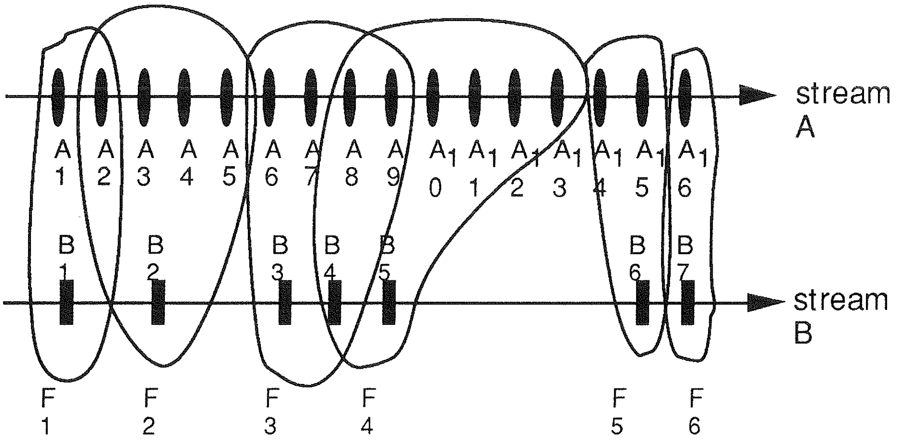

Figure 4 : coupling example, definition of a bundle

Let us consider a bundle build with two streams A and B. A coupling *L* describing the synchronization constraint between A and B can be defined in the following manner :

$F_k = L ((A_{k1}, A_{k2}, ..., A_{km}), (B_{k1}, B_{k2}, ..., B_{kn}))$ where $A_{k1}, A_{k2}, ..., A_{km}$ is a set of IU of the stream A and $B_{k1}, B_{k2}, ..., B_{kn}$ is a set of IU of th stream B.

Many definitions can be used to express a coupling (for example, by using causally dependances between the IUs [CCS93] [CRO94]).

We can consider a particular case of coupling where the IUs are coupled two by two : each slice F_k represents the UIs A_k and B_k : $\forall k$, $F_k = (A_k, B_k)$. The synchronization layer has to present the slices to the upper layer according to the synchronization constraints on the bundle.

4.2 Expression of the inter-stream synchronization

All the defintions given for the IU can be extended to the slices. A constraint C on a bundle F can be expressed by the constraint C on each F_k of the bundle F :

$$C(F) = <C(F_k)> \qquad (11)$$

We suppose that $C(F_k)$ represents the absolute presentation date that is required for the slice F_k. $R(F_k)$ is the reception date of the slice F_k and $P(F_k)$ is the effective presentation date of F_k.

The preservation of the temporal signature for the bundle F is defined by the following expression :

$$\forall\ k, P(F_k) = C(F_k) \qquad (12)$$

4.3. Characterization of the inter-stream synchronization loss

At first, it is necessary to verify if the bundle can be synchronized. The intra-stream constraints required on each stream of the bundle have to be compatible with the inter-stream constraint. The figure 5 shows the different cases for a bundle F composed of 2 streams A and B.

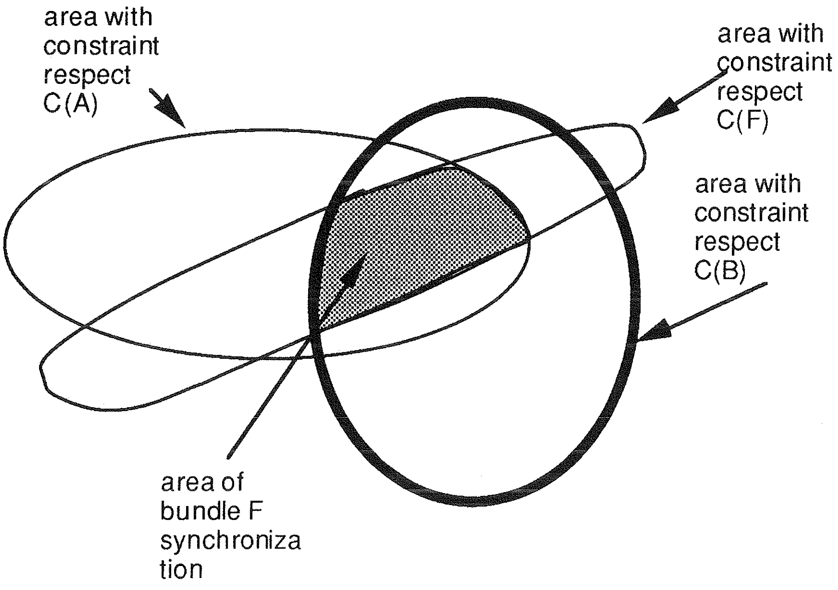

Figure 5 : synchronization capacity of a bundle

Taking in account C(A), C(B) and C(F), we have to be situated in the synchronizable area of F. To have a synchronizable bundle F(A,B), and with the hypothesis of one-to-one coupling and absolute date, C(A) must be equal to C(B). In this case, an inter-stream synchronization loss is necessary due to a synchronization loss of at least one an IU of one stream.

If we suppose that the presentation duration is 0, then the synchronization loss on a slice F_k has the following expression :

$$loss\_S(F_k, t) \Leftrightarrow \exists X \in \{ A_k, B_k \}/(\bar{r}(X)) \text{ and } (t>C(X)-\delta) \text{ or } (r(X)) \text{ and } (R(X)>C(X)-\delta) \quad (13)$$

Let us remark that if the synchronization loss is due to an A_k, then the constraint C(A) cannot be guaranteed. Several solutions exist to handle the stream B :

- (1) stop the stream B delivering.
- (2) continue the stream B delivering (respecting the intra-stream synchronization C(B)).

The first choice leads to an intra-stream synchronization loss on B. The second choice has no influence on the stream B, and has to be taken if no concealment mechanism isavailable.Consequently, concealment mechanisms are also needed for the bundle.

4.4 Non synchronization phase and inter-stream concealment mechanisms

As for a stream, let us define the presentation function Y of a bundle F, with a concealment mechanism. This function defines the slice to be delivered at the instant t :

$$Y(t) =: \text{if } t == C(F_k) \text{ then if } r(F_k) \text{ then } F_k \text{ else concealment\_action}(t). \quad (14)$$

Let us suppose that the bundle F is defined with a one-to-one coupling between the streams A and B, we can define the function "last-safe" that associates to t the last slice on which no synchronization loss has been observed.

$$\text{last-safe}(F,t) =: F_j \,/\, \overline{loss\_S}(F_j,t) \text{ and } ((\forall\ i>j \,/\, (C(F_i) < t), loss\_S(F_i, C(F_i)) \quad (15)$$

The mechanism (15) would not have influence on the intra-stream synchronizations if, at each intant t, we have :

$$(\text{last-safe}(A, t) \in \text{last-safe}(F, t)) \text{ and } (\text{last-safe}(B, t) \in \text{last-safe}(F, t)) \quad (16)$$

For example, if we suppose that the synchronization loss is due to the IU A_k of the stream A, then the function "last-safe" applied to the bundle (when we have to deliver $F_k = (A_k, B_k)$) gives $F_j = (A_j, B_j)$ with $A_j \neq A_k$ and $B_j \neq B_k$. Let us remark that if the function " last-safe" independantly applied on the streams A and B gives respectively A_j and B_k.

The condition (16) is not always verified. The choice (1) gives greater importance to the inter-stream synchronization, and the choice (2) gives greater importance to the preservation of the synchronization of the stream B.

The figure 6 shows what is delivered a the instant $t = C(F_k)$.

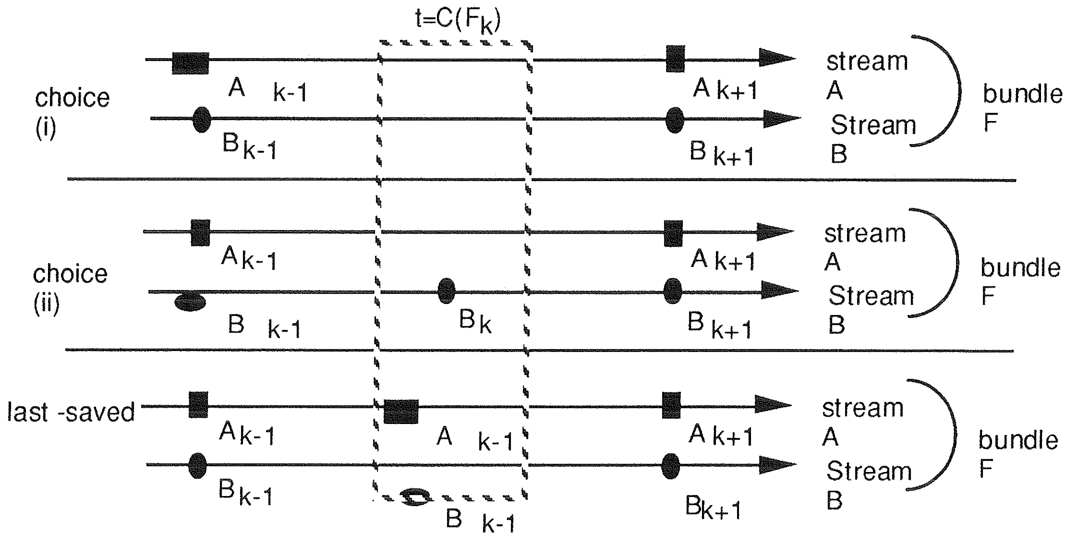

Figure 6 : concealment alternatives

Several others solutions exist. For example we can apply the function "last-safe" on the stream A, then deliver the IU A_{k-1} together with the IU B_k. In this case a slice is not delivered, but two IUs, and A_{k-1} and B_k are not bound in a slice (no one-to-one coupling).

All these possibilities show that the concealment mechanisms for a bundle are very complex and that more informations from the application are needed to provide "good" mechanisms according to the user needs.

5. Specification of the inter-stream concealment mechanisms

Several propositions can be found in the literature to achieve inter-stream synchronizations. In [CCO94], restricter algorithms are proposed that control the information unit delivery according to dependancy expressions. In [DSSS93], a set of relations between the streams of a bundle is proposed : master stream, logical OR between streams, logical AND between streams. For the formal definition of synchronizations, the model of TSPN (Timed Petri Nets) is used. With the relations coming from the application requirement, it is possible to build a concealment strategy to respect the relations and to deliver the streams of a bundle according to the defined relations.

We propose, as presented in [STE90], to express the concealment mechanisms using a programmation language. For example, let us propose 2 streams in a bundle (one stream of motion video, the other one of audio).

```
Stream A :
{full motion video}
display(experiment_video)
...
Synchronize with B at end
   mode(order_and_temporal)
while waiting
   display(start_image)
exception
    display(last_safe_image)
```

```
Stream B :
{audio)
play (experiment_audio)
...
Synchronize with A at end
   mode(order_and_temporal)
while waiting
    play(music (Bach))
exception
    play(silence)
```

We can remark that the synchronization requirements are coming from the application. If these requirements are not specified, some default options can be available according to the type of the stream.

In our example, we want to respect the order and the temporal signature of the two streams. The wainting action deals with the beginning of the streams before the first synchronization, and the exception action deals with the loss synchronization.

6. Conclusion

In this paper, we have studied the loss of synchronization in distributed multimedia application. To achieve synchronizations several steps have to be consider : maintain of the synchronization, synchronization loss detection, concealment mechanisms, synchronization recovery.

We have put to the fore the need of concealment mechanisms. These mechanisms are dedicated to mask the consequences of loss of synchronization. Intra-stream and inter-stream synchronization mechanisms have been presented for distributed multimedia application, using formal definitions. Then we have proposed one simple concealment mechanism : last_safe. Taking into account three different coding techniques, we show that the use of this concealment mechanism is useful for some but inopportune for others.

In further works, we will propose protocol mechanisms for all the above steps and we will take in account the human requirements and the types of the streams of the application to achieve "at best" synchronization in multimedia communication systems.

7. References

[ANH91] D.P. Anderson, G. Homsy. A Continuous Media Input/Output Server and its Synchronization Mechanism. IEEE Computer, Vol 8, n° 10, October 1991.

[CCS93] JP.Courtiat, L.Carmo, P.deSaqui-Sannes. Towards Multimedia Communication Services. 4th Workshop on Future Trends of Distributed Computing Systems. Lisbonne. September 1993.

[CCSBDH93] J. P. Courtiat, L. Carmo, P. de Saqui-Sannes, L. Besse, L. Dairaine, E. Horlait. Synchronisation multimedia : terminologie et concepts de base. Rapport du projet CESAME CNRS. Avril 1993.

[CRO94] J.P. Courtiat, L. Rust, R. Oliveira. A New Mechanism for Achieving Inter-Stream Synchronization in Multimedia Communication Systems. IEEE International Conference on Multimedia Computing and Systems. Boston, USA. May 1994.

[DIAZ92] M. Diaz. Document Préliminaire du Projet CESAME. LAAS CNRS. Toulouse. 1992.

[DSSS93] M. Diaz, D. Senac, P. de Saqui-Sannes. Un modèle pour la spécification et le contrôle de la synchronisation multimédia dans un environnement distribué. CFIP'93. Montréal - Canada. Septembre 1993.

[EDP92] J. Escobar, D. Deutsch, C. Partridge. Flow Synchronization Protocol. Research Report BBN, 10 Moulton Street, Cambridge, MA02138, USA. March 1992.

[H261] Recommandation H.261 : Video Codec for Audiovisual Services at p*64 kbits/s. CCITT, white book. 1990.

[JPEG] JPEG, ISO10918. Joint Photographics Expert Group.

[LIG91] T. Little, A. Ghafoor. Multimedia Synchronization for Broadband Integrates Services. IEEE Journal on Selected Areas in Communication, Vol 9, n° 9, Dec. 1990.

[MPEG] MPEG, ISO 11172. Moving Pictures Expert Group.

[MPEG2] Recommandation H.262 : Generic Coding of Moving Pictures and Associated Audio, MPEG. ISO/IEC 13818-2, November 1993.

[NIC90] C. Nicolaou. An Architecture for Real Time Multimedia Communication Systems. IEEE Journal on Selected Areas in Communication. Vol 8, n° 3, April 1990.

[PRY93] M. De Prycker. Asynchronous Transfer Mode. Ellys Horwood. 1993.

[RRV90] P.V. Rangan, S. Ramanthan, H.M. Vin, T. Kaeppner. Techniques for Multimedia Synchronization in Network File Systems. Computer Communication. Vol 16, n° 3, April 1990.

[SDF93] H. Santoso, L. Dairaine, S. Fdida, E. Horlait. Preserving Temporal Signature : A Way to convey Timed Constrained flows. GLOBECOM'93. IEEE Press.

[SEN93] R. Steinmetz, C. Engler. Human Perception of Mulimedia Synchronization. Research Report. IBM European Networking Center. n° 43.9310. 1993.

[STE90] R. Steinmetz. Synchronization Properties in Multimedia Systems. IEEE Journal on Selected Area in Com., Vol 8, n° 3, April 1990.

[VBLC94] G. Viho, B. Cousin, A. Leger, R. Castanet. Etude de la perte de synchronisation dans la transmission de données multimedia. Rapport du projet CESAME. CNRS. Mai 1994.