

IRISA
INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

PUBLICATION
INTERNE
N° 973

Masquage de la perte de synchronisation pour les applications multimédias réparties

Bernard COUSIN & Gagnon VIHO

Masquage de la perte de synchronisation pour les applications multimédias réparties*

B. Cousin et G. Viho

Programme 1 : Architectures parallèles, bases de données,
réseaux et systèmes distribués

Projet ADP et PAMPA

Publication interne n°973

Novembre 1995

20 pages

Résumé : Divers mécanismes ont été définis et sont actuellement utilisés pour le maintien de la synchronisation des flux multimédia. Toutefois, il arrive que malgré cela, on observe des pertes de synchronisation dues notamment à la gigue, à des interruptions inattendues du flux ou à la congestion sur le réseau de transmission. Généralement le choix alors effectué lors de la phase de désynchronisation, consiste à ignorer les données en provenance du réseau, rendant ainsi la perte de synchronisation perceptible. Dans cet rapport, nous étudions la perte de synchronisation lors de la transmission de données multimédia. Nous mettons en évidence la nécessité de développer des mécanismes destinés à masquer la perte de synchronisation au cours de la phase de désynchronisation. Nous caractérisons formellement la perte de synchronisation intra-flux et inter-flux et définissons un mécanisme de masquage : le *dernier_sauf*.

Mots-clé : synchronisation, multimédia, transmission, flux, faisceau, masquage.

* Ce travail a bénéficié du soutien de France Télécom/CNET au titre du marché n° 92 1B 178 - Projet CESAME : Conception formelle de systèmes coopératifs haut débit multimédia



Concealment of Synchronization Loss for Distributed Multimedia Applications

Abstract: Several mechanisms have been defined and are now used to maintain synchronization between multimedia flows. Meanwhile, losses of synchronization sometimes occur specially due to jitter, or unexpected interruption of the flow, or congestion in the network. Generally, data coming from the network during the desynchronization period are ignored. The loss of synchronization is then perceptible. In this report, we study the loss of synchronization in multimedia data transfer. We bring out the need of developing concealment mechanisms dedicated to mask the desynchronization period which follows the loss of synchronization. We characterize formally the intra-flow and inter-flow loss of synchronization and we define a concealment mechanism : the *dernier_sauf*.

Key-words: synchronization, multimedia, data transfer, flow, stream, concealment.

1 Introduction

L'ensemble des informations multimédia à transmettre à un instant donné peut être vu comme un *faisceau* de données composé de différents *flux* (flux d'images, flux sonores, flux de textes, etc.). Un des thèmes centraux dans le domaine de la transmission de données multimédias est la synchronisation des différents flux composant un faisceau. En effet, l'information élémentaire d'un flux doit souvent être présenté en même temps que des informations élémentaires d'autres flux du même faisceau (par exemple, image/son).

Divers mécanismes (utilisation de buffers, par exemple) sont utilisés pour le maintien de la synchronisation des flux. Toutefois, il arrive que malgré cela, on observe des pertes de synchronisation (saturation des buffers...) dues notamment à la gigue, à des interruptions inattendues du flux ou à la congestion sur le réseau de transmission. Généralement le choix alors effectué consiste à ignorer les données arrivant lors de la phase de désynchronisation (entre l'instant où l'on observe la perte de synchronisation et l'instant où a lieu la reprise).

Dans ce rapport, nous étudions l'influence de la perte de la synchronisation *intra-flux* (à l'intérieur de chacun des flux) et *inter-flux* (liant les flux d'un faisceau). Nous montrons l'influence de la mise en œuvre d'un mécanisme de masquage sur le flux et le faisceau. Pour finir, nous montrons la nécessité de disposer de moyens permettant de spécifier les contraintes de synchronisation intra-flux et inter-flux ainsi que les mécanismes de masquage de la désynchronisation.

2 Contexte de l'étude

Un flux peut être défini comme un ensemble d'unités d'information reliées entre-elles par des contraintes de synchronisation [1, 4]. Nous supposons qu'il existe un *service de transfert* de données chargé d'assurer le transport de flux entre plusieurs entités de niveau applicatifs, situées sur des sites distants. Or, la transmission d'un flux de données multimédias à travers un réseau asynchrone engendre des décalages temporels entre les unités d'information. Il est donc indispensable de mettre en œuvre au-dessus du service de transfert (et au-dessous du niveau application) des mécanismes pour préserver (vis-à-vis des entités de niveau application) les contraintes de

synchronisation des flux transmis [2, 3]. En termes de service, il s'agit de résoudre ce problème principalement vis-à-vis du faisceau présenté, c'est-à-dire qu'à partir du faisceau reçu, comment faire pour que le faisceau présenté (cf figure 1), respecte les contraintes de synchronisation imposées sur le faisceau ?

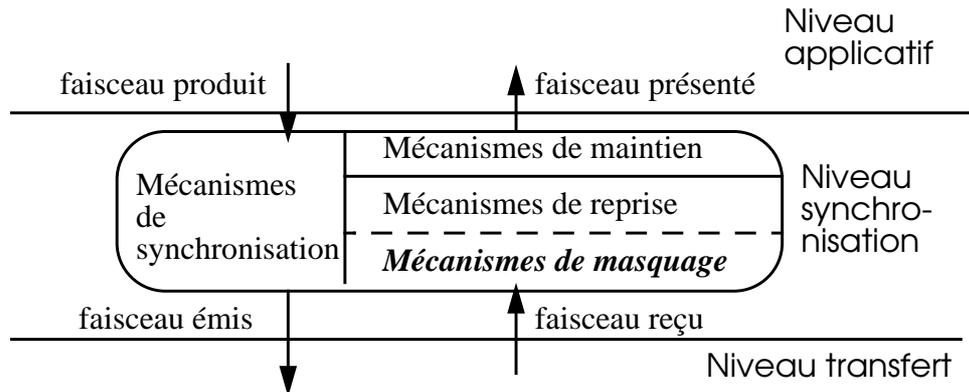


Figure 1 : Architecture de communication multimédia

Divers mécanismes ont été ainsi définis [2, 3, 4, 6, 17, 18] pour le maintien de la synchronisation des flux. Malgré cela, les phénomènes tels que la gigue, des interruptions inattendues du flux ou la congestion sur le réseau de transmission mettent parfois en défaut ces mécanismes. On observe alors une perte de synchronisation. La perte de synchronisation intra-flux peut être définie comme l'instant où l'on observe un non-respect des contraintes de synchronisation liant les unités d'information du flux, *ceci malgré les mécanismes mis en œuvre pour le maintien de la synchronisation*. De même, la perte de synchronisation inter-flux est l'instant où l'on observe le non-respect des contraintes de synchronisation liant les flux du faisceau.

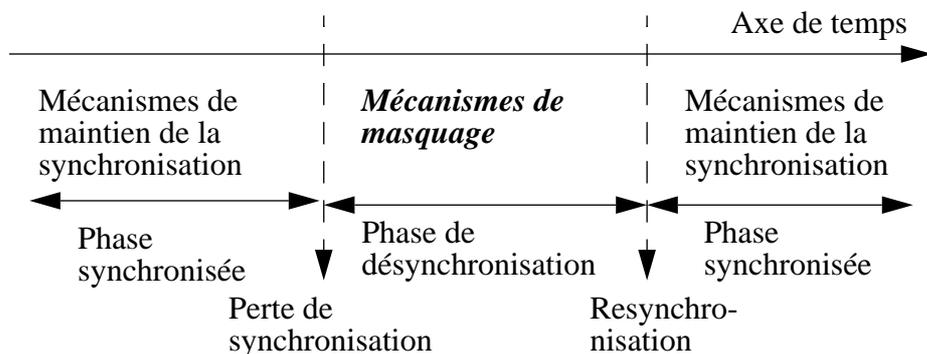


Figure 2 : Vie d'un faisceau et mécanismes de masquage de la désynchronisation

On doit alors mettre en œuvre des mécanismes pour effectuer la reprise de synchronisation mais également des *mécanismes de masquage* destinés à masquer la désynchronisation dans l'intervalle de temps séparant la perte de synchronisation et la reprise de synchronisation (cf figure 2).

Nous n'envisageons ici que l'étude des transmissions point à point de flux d'un faisceau et de leur synchronisation. Nous caractérisons ensuite formellement la perte de synchronisation intra-flux et inter-flux. Nous montrons l'influence de la mise en œuvre d'un mécanisme de masquage sur le flux et le faisceau. Pour finir, nous montrons la nécessité de disposer de moyens permettant de spécifier les contraintes de synchronisation et les mécanismes de masquage de la perte de synchronisation.

3 Services de synchronisation intra-flux et caractérisation de la perte de synchronisation intra-flux

Les contraintes de synchronisation liant les unités d'information d'un flux sont généralement définies en termes de préservation d'une certaine *signature temporelle* sur les unités d'information. Ainsi, respecter les contraintes imposées sur un flux revient à préserver sa signature temporelle [1]. Par exemple, si on considère qu'une unité d'information est une image entière dans un flux d'images animées, alors la préservation de la signature temporelle peut s'exprimer comme suit : "une image doit être présentée toutes les 25^{ème} de seconde".

La notion d'unité d'information dépend non seulement du type de média [4] mais également du niveau de synchronisation exigé sur le média. Nous noterons A_k , la $k^{\text{ème}}$ unité d'information d'un flux A . Le flux A est alors la suite formée par l'ensemble de ses unités d'information, noté de la manière suivante : $A = \langle A_k \rangle$.

3.1 Expression des contraintes de synchronisation intra-flux

Un *flux contraint* est un flux muni d'une signature temporelle. Une signature temporelle C souhaitée sur un flux A peut se traduire par la signature temporelle souhaitée pour chaque unité d'information A_k constitutive du flux. Nous exprimons cela de la manière suivante :

$$C(A) = \langle C(A_k) \rangle \quad (1)$$

Dans l'expression précédente $C(A_k)$ peut désigner la date absolue de présentation souhaitée pour l'unité d'information A_k , ou la date relative par rapport à une unité d'information quelconque A_j , ou encore toute autre

expression (sous la forme de dépendance causale [14], par exemple) décrivant la contrainte à imposer sur A_k pour respecter la signature temporelle C sur l'ensemble du flux A . Dans un premier temps, nous supposons que $C(A_k)$ désigne une date absolue. Ainsi par exemple, la préservation de la signature temporelle sur un flux A peut s'exprimer par la date à laquelle on devrait présenter les unités d'information : $C(A_k)$ désigne alors cette date pour l'unité d'information A_k du flux A . Si nous notons $P(A_k)$ la date de présentation effective de l'unité d'information A_k , alors la *préservation de la signature temporelle* sur le flux A s'exprime de la manière suivante :

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

Nous notons $R(A_k)$, la date de réception effective de l'unité d'information A_k . De façon générale, la date de présentation $P(A_k)$ dépend de la date de réception $R(A_k)$. En effet, le système chargé de la présentation des unités d'information introduit un *délai de présentation* qui varie au cours du temps. Toujours pour simplifier notre propos, nous supposons que ce *délai de présentation* est constant et égal à δ . Cela correspond à une garantie d'un temps de réponse constant offerte par le système de présentation et se traduit par : $P(A_k) = R(A_k) + \delta, \forall k$.

La condition (2) serait systématiquement satisfaite si l'on suppose que le délai de transmission est constante, c'est à dire que le réseau de transport offrirait alors un service synchrone. Or, cette hypothèse est peu réaliste car le délai de transmission dans les réseaux informatiques varie au cours du temps avec la charge du réseau. Cette variabilité peut être due à des phénomènes tels que, les interruptions inattendues du flux ou la congestion sur le réseau de transmission,... On observe alors une perte de synchronisation.

3.2 Caractérisation de la perte de synchronisation intra-flux

Pour une unité d'information A_k d'un flux A , $r(A_k)$ signifiera que l'on a reçu A_k et $\bar{r}(A_k)$ indique la non-réception de A_k . Remarquons que si l'on a $r(A_k)$ alors on a une valeur pour la date effective de réception $R(A_k)$ de l'unité d'information A_k .

On observe *une perte de synchronisation intra-flux* à un instant t sur une unité d'information A_k , notée par $per_S(A_k, t)$ si et seulement si :

$$\begin{aligned} per_S(A_k, t) \iff & (\bar{r}(A_k) \text{ et } (t > C(A_k) - \delta)) \\ & \text{ou } (r(A_k) \text{ et } (R(A_k) > C(A_k) - \delta)) \end{aligned} \quad (3)$$

Ceci correspond à une détection instantanée de la perte de synchronisation intra-flux. La première partie de l'expression précédente signifie que l'on n'a pas reçu A_k et que la date de présentation $C(A_k)$ souhaitée pour A_k

est dépassée. La deuxième partie signifie que l'on a reçu A_k mais plus tard que la date de présentation souhaitée pour A_k (permettant de préserver la contrainte globale $C(A)$ imposée sur le flux).

Une remarque s'impose ici pour dire que quand on parle de perte de synchronisation de façon générale, il s'agit de la détection a posteriori (après le déroulement du flux). L'écriture de cette propriété ne fait alors plus intervenir l'instant de la détection et s'écrit alors comme suit :

$$perte_S(A) \Leftrightarrow \exists A_k / (\bar{r}(A_k)) \text{ ou } (r(A_k) \text{ et } R(A_k) > C(A_k) - \delta) \quad (4)$$

Ceci veut dire que l'on n'a jamais reçu A_k ou qu'on l'a reçu plus tard que sa date de présentation souhaitée.

Bien que n'étant généralement pas considérée comme une contrainte de synchronisation, la préservation de l'ordre peut être exigé d'un flux sur lequel on impose aucune contrainte temporelle. C'est l'exemple d'une séance de projection muette de diapositives où l'on exige uniquement que les diapositives arrivent dans l'ordre.

Dans tous les cas, la *préservation de l'ordre* revient à dire qu'une unité d'information A_k doit être présentée avant A_{k+1} , quelque soit k . Cela s'exprime aisément de la manière suivante :

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

Lorsqu'une perte de synchronisation est observée, il faut mettre en œuvre des mécanismes pour la reprise de synchronisation. L'attitude généralement adoptée consiste alors à ne pas tenir compte des unités d'information arrivant dans l'intervalle séparant l'instant où la perte de synchronisation intervient et l'instant où aura lieu la reprise de synchronisation. Or, il est possible de gérer en parallèle avec la reprise et de manière judicieuse (en tenant compte des contraintes de synchronisation) cette *phase de désynchronisation*. L'objectif à atteindre ici étant de masquer la désynchronisation en offrant un service de synchronisation qui se rapproche le plus possible du service global initial. C'est la raison pour laquelle, nous appelons les mécanismes mis en œuvre à cet effet, les *mécanismes de masquage* de la perte de synchronisation.

3.3 La phase de désynchronisation intra-flux et mécanismes de masquage

Nous définissons la fonction de présentation X qui à un instant t donné, associe l'unité d'information d'un flux présenté à cet instant. Une fonction de présentation qui préserve la contrainte C sur un flux A vérifie alors $X(C(A_k)) = A_k, \forall k$. La fonction de présentation peut s'écrire formelle-

ment de la manière suivante :

$$X(t) = \begin{cases} \text{si } t = C(A_k) \\ \text{alors si } r(A_k) \text{ alors } A_k \text{ fsi} \\ \text{fsi} \end{cases} \quad (6)$$

Lorsqu'on ne dispose pas de mécanismes de masquage ni de reprise de synchronisation, la fonction X décrit les mécanismes de maintien de la synchronisation. L'expression (6) signifie que l'on présente uniquement les unités d'information que l'on reçoit à temps. Elle représente une situation "idéale", où les unités d'information sont reçues à temps. Ainsi, si on observe une perte de synchronisation à un instant t , on n'a pas de définition associée à $X(t)$. Il est donc nécessaire de prévoir les cas de perte de synchronisation dans la fonction de présentation. C'est le rôle des mécanismes de masquage qui va donc consister à associer une valeur (c'est à dire une unité d'information) à $X(C(A_k))$ lorsque l'on n'a pas reçu A_k à temps. Dans ce cas, la fonction de présentation s'écrit :

$$X(t) = \begin{cases} \text{si } t = C(A_k) \\ \text{alors si } r(A_k) \text{ alors } A_k \\ \text{sinon } < \text{masquage}(t) > \\ \text{fsi} \\ \text{fsi} \end{cases} \quad (7)$$

Différents mécanismes de masquage peuvent être envisagés, par exemple en présentant la dernière unité d'information que l'on a correctement reçue, ou encore en présentant une interpolation de cette dernière, etc.

Considérons le mécanisme de masquage qui consiste à présenter l'unité d'information précédente. La fonction de présentation s'écrirait alors :

$$X(t) = \begin{cases} \text{si } t = C(A_k) \\ \text{alors si } r(A_k) \text{ alors } A_k \\ \text{sinon } A_{k-1} \\ \text{fsi} \\ \text{fsi} \end{cases} \quad (7\text{bis})$$

Malheureusement, on s'aperçoit que ce mécanisme de masquage est particulièrement frustré et peu efficace. En effet, l'unité d'information précédente n'est pas forcément reçue. Il convient donc de raffiner un peu plus le mécanisme de masquage par exemple pour récupérer la dernière unité d'information correctement reçue et présentée. Pour ce faire nous définissons

la fonction *dernier_sauf* qui à un instant t , associe la dernière unité d'information A_j sur laquelle, on n'a pas observé une perte de synchronisation. Cette fonction s'écrit formellement de la manière suivante :

$$dernier_sauf(A, t) = A_j / \overline{perte_S}(A_j, t) \text{ et} \\ ((\forall i > j / (C(A_i) < t), perte_S(A_i, C(A_i)))) \quad (8)$$

Une fonction de présentation qui utilise le mécanisme de masquage présentant la dernière unité d'information correctement reçue s'écrit alors de la manière suivante :

$$X(t) = \begin{array}{l} \text{si } t = C(A_k) \\ \quad \text{alors si } r(A_k) \text{ alors } A_k \\ \quad \quad \quad \text{sinon } dernier_sauf(A, t) \\ \quad \quad \quad \text{fsi} \\ \text{fsi} \end{array} \quad (9)$$

Par la suite, nous appellerons ce mécanisme de masquage le *dernier_sauf*.

Bien que moins frustré que le mécanisme de masquage défini en (7bis), on s'aperçoit que l'efficacité du *dernier_sauf* peut dépendre du type de codage utilisé pour le flux. Pour illustrer notre propos, considérons les trois types de codage suivants sur un flux d'images :

- *codage indépendant* (du type M-JPEG[11]), où le codage/décodage des unités d'information s'effectue indépendamment l'une de l'autre. L'unité d'information est notée ici I (pour codage indépendant). Le flux est alors constitué uniquement d'images I.

Si l'on observe une perte de synchronisation sur une unité d'information, alors le *dernier_sauf* correspond à l'image précédemment présentée.

Notons ici que sur un flux sonore, l'effet du *dernier_sauf* est plus perceptible que sur un flux d'images. En effet, lorsqu'il y a une perte de synchronisation sur un flux sonore et qu'on ne dispose pas de mécanismes de masquage, cela se traduit par un son haché.

- *codage différentiel unidirectionnel* (du type H261[12]), où le codage/décodage d'une unité d'information peut dépendre de la précédente unité d'information. Dans ce cas, l'unité d'information est notée P.

S'il y a une perte de synchronisation sur une quelconque des unités d'information, alors on présente (comme dans le cas précédent) la précédente (I ou P) correctement présentée. Par contre, la reprise de synchronisation ne peut se faire que sur une unité d'information de type I.

- *codage différentiel bidirectionnel* (du type MPEG[13]), où le codage/décodage d'une unité d'information peut dépendre d'une unité d'information précédente et/ou suivante. Aux deux autres types d'unité d'information précédents, se rajoute celle que l'on note B dont le codage dépend à la fois d'une unité d'information précédente et future de type I ou P.

En cas de perte de synchronisation sur des unités d'information de type I ou P, on a le même traitement que dans le codage précédent. Par contre, si on observe une perte de synchronisation sur unité d'information de type B, le *dernier_sauf* ne peut être qu'une unité d'information de type I ou P. En effet, comme par définition une unité d'information de type B dépend à la fois d'une unité d'information précédente et future, une perte de synchronisation sur une unité d'information de type B ne peut provenir que de la non-réception (pas à temps) de l'unité d'information future. Donc, la perte de synchronisation a lieu sur la première unité d'information de type B (après une unité d'information de type I ou P). Le *dernier_sauf* va donc consister à présenter la précédente unité d'information de type I ou P.

L'inconvénient dans ce cas étant alors que l'unité d'information ainsi présentée soit "ancienne" dans le déroulement du flux. C'est le cas par exemple si l'intervalle de temps séparant deux unités d'information de type I ou P est "assez long". Dans MPEG-vidéo, un paramètre permet de fixer le nombre M d'unités d'information de type B. L'efficacité du mécanisme du *dernier_sauf* est inversement proportionnelle à la valeur de M.

Nous avons montré dans ce paragraphe, l'intérêt que peut avoir le mécanisme de masquage dans la présentation d'un flux pour masquer la perte de synchronisation intra-flux. Or dans les applications multimédias, un flux est toujours combiné à d'autres flux (formant ainsi un faisceau) avec des contraintes de synchronisation inter-flux liant les unités d'information de différents flux. Il faut donc mettre en œuvre des mécanismes de masquage inter-flux destinés à masquer la perte de synchronisation sur un faisceau.

4 Services de synchronisation inter-flux et mécanismes de masquage inter-flux

L'ensemble des informations multimédias à transmettre peut être vu comme un *faisceau* de données composé de différents *flux* (flux d'images, flux sonore, flux de textes, ...). C'est le rôle du système chargé de la présentation des données, d'assurer le service de synchronisation des faisceaux exigée par les entités-applications. Il va donc s'agir de présenter l'ensemble des unités d'information qui composent les différents flux en respectant les con-

traintes de synchronisation liant ces flux du faisceau. Pour ce faire, nous avons besoin d'identifier l'objet élémentaire du faisceau. Il nous faut donc introduire pour le faisceau une notion équivalente à celle des unités d'information pour le flux.

4.1 Définition d'un faisceau - notion de couplage et de tranches

Un *couplage* exprime les contraintes de synchronisation liant des unités d'information des différents flux composant un faisceau. Tout ensemble d'unités d'information de différents flux associés par un couplage est appelé *tranche* du faisceau. Un faisceau est alors totalement défini par la suite de tranches qui la composent.

De manière analogue à l'unité d'information pour le flux, une tranche constitue un objet élémentaire du faisceau. Nous noterons F_k la $k^{\text{ème}}$ tranche d'un faisceau F . Un faisceau F sera alors noté :

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

Considérons un faisceau F composé de deux flux A et B . Un couplage \mathcal{L} décrivant la contrainte de synchronisation entre A et B peut être défini comme suit :

$F_k = \mathcal{L}((A_{k1}, A_{k2}, \dots, A_{km}), (B_{k1}, B_{k2}, \dots, B_{kn}))$ où $A_{k1}, A_{k2}, \dots, A_{km}$ est un ensemble d'unités d'information du flux A et $B_{k1}, B_{k2}, \dots, B_{kn}$ un ensemble d'unités d'information du flux B .

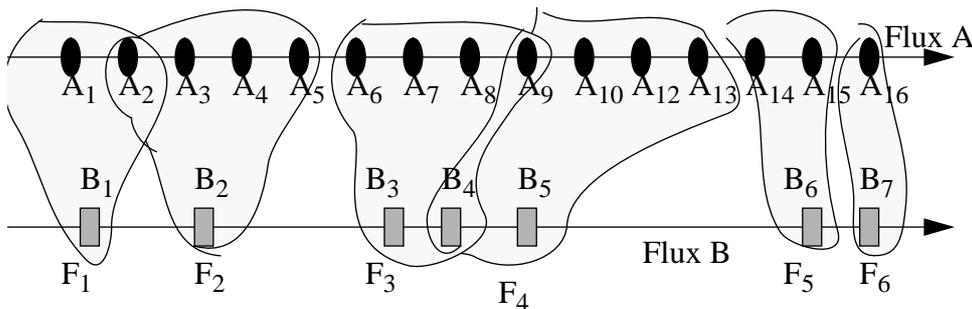


Figure 3 : Un exemple de couplage - Définition d'un faisceau

Il existe plusieurs manières de définir un couplage. On peut l'exprimer par exemple sous la forme de dépendances causales entre les unités d'information [5]. Dans un premier temps, nous considérons un cas particulier de couplage où les unités d'information sont reliées deux à deux (par un couplage biunivoque), c'est à dire que la tranche F_k est composé de l'unité d'information A_k du flux A et de l'unité d'information B_k du flux B . Nous notons cela de la manière suivante : $F_k = (A_k, B_k), \forall k$.

Le rôle du système chargé de la synchronisation consiste alors à présenter les tranches du faisceau à l'entité de niveau supérieur en respectant les contraintes de synchronisation imposées sur le faisceau.

4.2 Expression de contraintes de synchronisation inter-flux

Compte tenue de la définition des tranches, toutes les définitions et notions temporelles définies pour les unités d'information s'appliquent également pour les tranches. Ainsi par exemple, un *faisceau contraint* est un faisceau muni d'une signature temporelle. Une signature temporelle C sur un faisceau F peut se traduire par la signature C sur chaque tranche F_k constitutive du faisceau. Nous exprimons cela de la manière suivante :

$$C(F) = \langle C(F_k) \rangle \quad (11)$$

Nous supposons (comme pour les flux) que $C(F_k)$ représente la date absolue de présentation exigée sur la tranche F_k . De manière analogue, $R(F_k)$ désignera la date de réception de la tranche F_k et $P(F_k)$ désignera la date de présentation effective de la tranche F_k .

La *préservation de la signature temporelle* du faisceau F s'exprime alors de la manière suivante :

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

4.3 Caractérisation de la perte de synchronisation inter-flux

Avant de caractériser la perte de synchronisation, il faut s'assurer que le faisceau est synchronisable. Il faut d'abord s'assurer que les contraintes (intra-flux) imposées sur chacun des flux soient compatibles avec la contrainte (inter-flux) sur le faisceau. En effet, pour un faisceau F composé de deux flux quelconques A et B , la figure 4 suivante montre les différents cas que l'on peut avoir.

En d'autres termes, il faut garantir qu'avec $C(A)$, $C(B)$ et $C(F)$, on est dans le domaine de synchronisabilité de F . Ainsi par exemple, pour que le faisceau $F = (A, B)$ soit synchronisable, nos hypothèses (de couplage biunivoque et de dates absolues) imposent que l'on ait : $C(A) = C(B)$. C'est à dire, il faut que : $\forall k, C(A_k) = C(B_k)$.

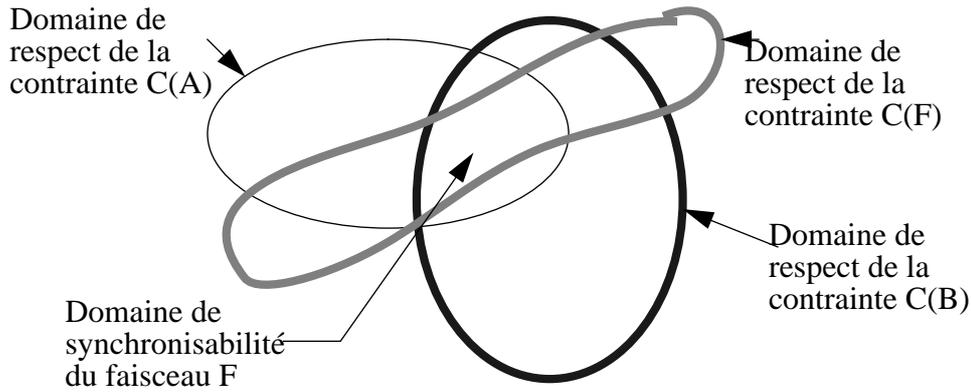


Figure 4 : Synchronisabilité d'un faisceau

Nous supposons par la suite que le faisceau est synchronisable. Nous considérons toujours qu'une tranche est composée de deux unités d'information des deux flux réunies par un couplage biunivoque. Dans ce cas-ci, il est évident que la perte de synchronisation inter-flux est nécessairement due à une perte de synchronisation concernant au moins une unité d'information d'au moins un des flux. Comme pour le flux et toujours pour simplifier, nous supposons que le délai de présentation est constant. La perte de synchronisation sur une tranche F_k s'exprime alors de la manière suivante :

$$perte_S(F_k, t) \Leftrightarrow \exists X \in \{A_k, B_k\} / (\bar{r}(X) \text{ et } (t > C(X) - \delta)) \\ \text{ou } (r(X) \text{ et } (R(X) > C(X) - \delta)) \quad (13)$$

Remarquons que cette perte de synchronisation influe non seulement sur le comportement général du faisceau mais également sur la synchronisation intra-flux de chacun des flux. En effet, si cette perte de synchronisation inter-flux (du faisceau) est due à A_k , alors il est évident que le flux A perd également sa synchronisation intra-flux : on ne peut plus garantir la contrainte $C(A)$. Comme de toute façon la synchronisation inter-flux ne peut plus être préservée (du moins, jusqu'à la reprise), on a plusieurs solutions pour le flux B, parmi lesquelles :

- (i) arrêter la présentation du flux B.
- (ii) continuer la présentation du flux B (respectant ainsi la synchronisation intra-flux $C(B)$).

Le choix (i) entraîne une perte de synchronisation intra-flux sur B alors que le choix (ii) n'influe pas sur le flux B. Par contre, le fait de continuer à présenter le flux B dans le choix (ii) est déjà en soi un mécanisme de masquage de la désynchronisation sur le faisceau.

Malgré nos hypothèses (dates absolues, couplage biunivoque), d'autres choix dépendant des objectifs de présentation visés peuvent être effectués. Ceci montre la nécessité de disposer de mécanismes de masquage non seulement sur les différents flux constituant le faisceau mais également sur le faisceau. Ces mécanismes de masquage sur le faisceau se traduisent par des actions sur les flux composant le faisceau, et peuvent de ce fait être différents des mécanismes de masquage sur un flux particulier lorsqu'il est traité indépendamment. En l'occurrence, il se peut que le *dernier_sauf* convienne à un flux lorsqu'il est traité de manière indépendante mais ne convienne pas lorsque ce flux est combiné avec un autre à travers un couplage pour former un faisceau. De manière analogue au flux, il faut donc également spécifier le mécanisme de masquage du faisceau.

4.4 La phase de désynchronisation inter-flux et mécanismes de masquage inter-flux

Comme nous l'avons fait pour un flux, nous pouvons définir la fonction de présentation Y d'un faisceau F intégrant un mécanisme de masquage. Cette fonction définit alors la tranche que l'on doit présenter à un instant donné. Elle s'écrit de la manière suivante :

$$Y(t) = \begin{cases} \text{si } t = C(F_k) \\ \quad \text{alors si } r(F_k) \text{ alors } F_k \\ \quad \quad \quad \text{sinon } \langle \text{masquage}(t) \rangle \\ \text{fsi} \end{cases} \quad (14)$$

Supposons toujours que le faisceau F est défini par un couplage biunivoque entre deux flux A et B et que nous voulions appliquer le mécanisme du *dernier_sauf* au faisceau F. Nous devons définir la fonction (également appelée *dernier_sauf*) qui à un instant t, associe la dernière tranche F_j sur laquelle, on n'a pas observé une perte de synchronisation. Cette fonction s'écrirait formellement de la manière suivante

$$\text{dernier_sauf}(F, t) = F_j / \overline{\text{perte_S}(F_j, t)} \text{ et} \\ ((\forall i > j / (C(F_i) < t), \text{perte_S}(F_i, C(F_i)))) \quad (15)$$

Le mécanisme du *dernier_sauf* sur le faisceau n'influerait pas sur les synchronisations intra-flux si à tout instant t , on a :

$$\begin{aligned} \text{dernier_sauf}(A, t) &= \text{dernier_sauf}(F, t)/A \text{ et} \\ \text{dernier_sauf}(B, t) &= \text{dernier_sauf}(F, t)/B \end{aligned} \quad (16)$$

où F_k/A (resp. F_k/B) dénote la projection de la tranche F_k sur le flux A (resp. B).

Or, si on reprend par exemple l'hypothèse que la perte de synchronisation inter-flux est causée par l'unité d'information A_k du flux A , alors le *dernier_sauf* appliqué au faisceau à l'instant $t=C(F_k)$ (l'instant où l'on doit normalement présenter $F_k = (A_k, B_k)$) implique la présentation de : $\text{dernier_sauf}(F, C(F_k)) = F_j = (A_j, B_j)$, avec $A_j \neq A_k$ et $B_j \neq B_k$. Alors que, le *dernier_sauf* intra-flux appliqué indépendamment au même instant $t=C(F_k)$ aux flux A et B donne :

$$\text{dernier_sauf}(A, C(F_k)) = A_j \text{ et } \text{dernier_sauf}(B, C(F_k)) = B_k.$$

La condition (16) n'est donc pas vérifiée. Par ailleurs, le *dernier_sauf* inter-flux (sur le faisceau) dans ce cas, fait alors présenter une unité d'information "ancienne" du flux B alors que l'on dispose de B_k . Le choix (i) du paragraphe précédent respecte la synchronisation inter-flux (couplage bi-univoque) alors que le choix (ii) privilégie la préservation de la synchronisation intra-flux B . En supposant que $j=k-1$, la figure 5 ci-après montre ce qui est présentée à l'instant $t=C(F_k)$ dans les 3 cas précédents.

On peut également choisir d'appliquer le *dernier_sauf* sur le flux A et présenter ainsi l'unité d'information A_{k-1} avec l'unité d'information B_k . Dans ce dernier cas, on ne présente pas une tranche mais deux unités d'information de deux tranches distinctes, puisque A_{k-1} et B_k ne peuvent pas être réunies par le couplage biunivoque (cf définition du paragraphe 4.1).

Toutes ces possibilités montrent que les mécanismes de masquage inter-flux peuvent être plus complexes que les mécanismes de masquage intra-flux et qu'il faut des informations complémentaires dans la spécification des mécanismes de masquage inter-flux.

Un type d'information supplémentaire qui permettrait de raffiner le mécanisme de masquage serait par exemple de définir une sorte de priorité sur les différents flux intervenant dans la composition du faisceau. Ils existent plusieurs modèles permettant d'exprimer cette priorité [14, 15] : dépendances causales, notion de flux maître ou esclave, lien ET_maître, etc...

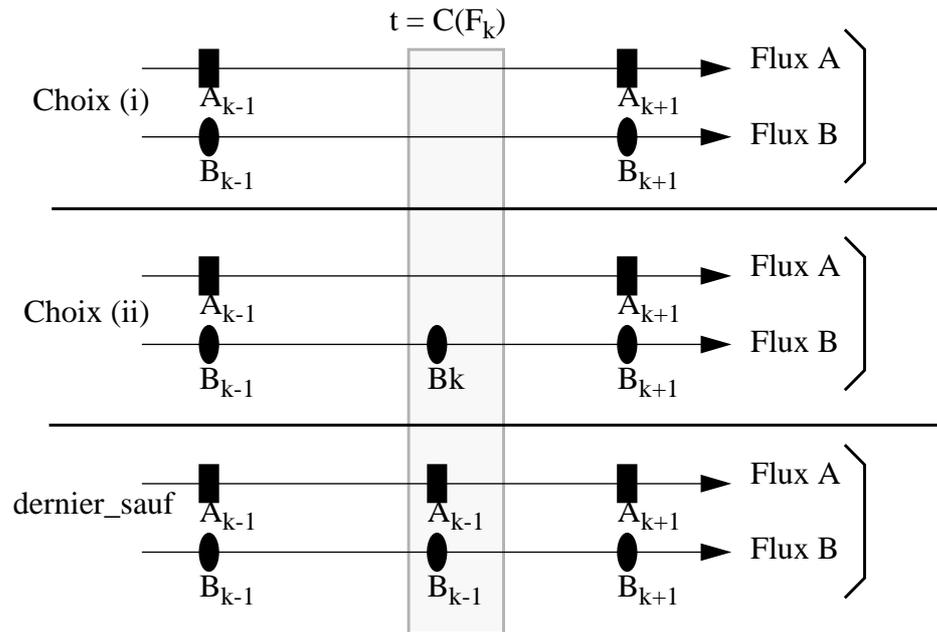


Figure 5: Conséquences des choix de masquage de la désynchronisation

Quelque soit le moyen utilisé pour exprimer cette priorité, il convient de préciser quels mécanismes de masquage on souhaite mettre en œuvre sur un flux donné, lorsqu'il y a une perte de synchronisation inter-flux, indépendamment de ceux utilisés dans le cas de la perte de synchronisation intra-flux. Il appartient ensuite au système chargé de la synchronisation de vérifier la compatibilité [8,16] de ces différents mécanismes.

4.5 Spécification des mécanismes de masquage inter-flux

Dans [4], Steinmetz propose un ensemble de primitives permettant de spécifier les mécanismes de masquage (ce qu'il appelle les *exceptions*). En utilisant ces primitives, une spécification des mécanismes de masquage entre deux flux A et B synchronisés par un couplage bi-univoque (appelé synchronisation *ET_maitre* dans [15]), pourrait s'écrire de la manière suivante :

```
Flux A :                               Flux B :
-- full-motion video                   -- audio

begin                                   begin
display(video)                          play(audio)
....                                    .....
SYNCHRONIZE WITH B AT end               SYNCHRONIZE WITH A AT end
    MODE ET_maître                       MODE ET_maître
WHILE WAITING                           WHILE WAITING
    display_last_image                    play(music_BACH)
EXCEPTION                                EXCEPTION
    dernier_sauf                          dernier_sauf
.....                                    .....
end;                                     end;
```

Dans cette spécification nous n'avons indiqué que les parties nécessaires au masquage inter-flux. Elle correspond par exemple au cas où, pendant la phase de désynchronisation, on opte pour le mécanisme du *dernier_sauf* pour la vidéo (flux A) et l'audio réel (flux B). Les primitives *display_last_image* et *dernier_sauf* sont supposées connues et implémentées dans le système chargé de la présentation des unités d'information. Avec une telle spécification, c'est ensuite le rôle du système chargé de la synchronisation de vérifier la compatibilité des mécanismes de masquage avec les contraintes inter-flux sur les flux A et B.

5 Conclusion

Dans ce rapport, nous avons étudié la perte de synchronisation dans la transmission de données multimédia. Nous avons mis en évidence la pos-

sibilité de mettre en œuvre des mécanismes de masquage destinés à masquer la phase de désynchronisation qui suit la perte de synchronisation. Nous avons caractérisé formellement la perte de synchronisation intra-flux et inter-flux et avons défini un mécanisme simple de masquage : le *dernier_sauf*. En considérant trois différents types de codages, nous avons montré que l'utilisation d'un tel mécanisme peut s'avérer intéressant pour un codage donné et inopportun pour d'autres.

Nous avons également montré que la réalisation du masquage inter-flux exigeait une spécification fine des contraintes de synchronisation entre les flux. Par ailleurs, nous avons montré qu'il est important pour le système de chargé de la synchronisation de savoir quel mécanisme de masquage il doit appliquer sur chacun des flux du faisceau. Ce qui nous a conduit à donner une manière de spécifier ces mécanismes.

Dans la suite de nos travaux, nous comptons généraliser l'étude de perte de synchronisation inter-flux en considérant des couplages plus lâches que le couplage biunivoque considéré dans ce rapport. Nous relierons ces couplages aux règles de tir du formalisme TSPN[15] et aux dépendances causales[14], en supprimant l'hypothèse de date absolue et de temps de traitement constant. Ceci nous donnera une caractérisation plus générale de la perte de synchronisation. Ainsi, nous pourrions définir d'autres mécanismes de masquage et leur implémentation dans quelques outils de transmission multimédia nous permettra de tester leurs efficacités.

Bibliographie

- [1] J.P. Courtiat, L.F.R.C. Carmo, P. de Saqui-Sannes, L. Besse, L. Dairaine & E. Horlait, "*Synchronisation multimédia : terminologie et concepts de base*", Rapport de projet Césame, Avril 1993.
- [2] P.V. Rangan, S. Ramanathan, H. M. Vin and T. Kaepfner, "*Techniques for Multimedia Synchronization in Network File Systems*", Computer Communication, vol. 16, no 3, April 1990.
- [3] H. Santoso, L. Dairaine, S. Fdida and E. Horlait, "*Preserving Temporal Signature : a Way to Convey Time Constrained Flows*", Proceedings of GLOBECOM'93, IEEE Press.
- [4] R. Steinmetz, "*Synchronization Properties in Multimedia Systems*", IEEE Journal on Selected Areas in Com., vol. 8, no 3, April 1990.
- [5] J.P. Courtiat, L.F.R.C. Carmo and P. de Saqui-Sannes, "*Towards Multimedia Communication Services*", Proceedings of fourth Workshop on Future Trends of Distributed Computing Systems, Lisbonne, September 1993.

- [6] D.P. Anderson, G. Homsy, “*A Continuous Media Input/Output Server and its Synchronization Mechanism*”, IEEE Computer, Vol 24, no 10, Oct. 1991
- [7] C. Nicolaou, “*An Architecture for Real-Time Multimedia Communication Systems*”, IEEE Journal on Selected Areas in Communication, Vol 8, no 3, April 1990.
- [8] T. Little & A. Ghafoor, “*Multimedia Synchronization Protocols for Broadband Integrated Services*”, IEEE Journal on Selected Areas in Communication, Vol 9, no 9, December 1991.
- [9] J. Escobar, D. Deutsch and C. Partridge, “*Flow Synchronization Protocol*”, Research Report BBN, 10 Moulton Street, Cambridge, MA02138, USA, March 1992.
- [10] R. Steinmetz and C. Engler, “*Human Perception of media Synchronization*”, Research Report, IBM European Networking Center, N° 43.9310 1993.
- [11] G.K. Wallace, “*The JPEG Still Picture Compression standard*”, Communications of the ACM, vol 34, pp 31-43, April 1991.
- [12] CCITT, “*Recommendation H.261 : Video codec for audiovisual services at p*64 kbits/s*”, CCITT white book, 1990.
- [13] “*Recommendation H. 262 : Generic Coding of Moving pictures and Associated Audio, MPEG*”, ISO/IEC 13818-2, November 1993.
- [14] L.F.R.C. Carmo, J.P. Courtiat & P. de Saqui-Sannes, “*Implementing Inter-Stream Synchronization by means of Conditionnal Dependency Expressions*”, LAAS Technical Report, October 1993.
- [15] M. Diaz, P. Sénac & P. de Saqui-Sannes, “*Synchronisation et objets multimédias*”, Rapport Césame, Juillet 1993.
- [16] T. Bloom, “*Evaluating synchronization mechanisms*”, in Proc. 7th Symp. Oper. Syst. principles, December 10-12, pp 24-32, 1979, Pacific Grove, CA.
- [17] CCETT, “*Multimedia Synchronization*”, CCETT Int. note : AFNOR adhoc group on AVI standardization, July 1988.
- [18] S.R. Faulk & D. L. Parnas, “*On synchronization in hard-real-time systems*”, Comm. ACM, vol. 31, n° 3, pp. 274-287, March 1988.

