



HAL
open science

Conception et formalisation d'une application de visioconférence coopérative. Application et extension pour la téléformation

Philippe Owezarski

► **To cite this version:**

Philippe Owezarski. Conception et formalisation d'une application de visioconférence coopérative. Application et extension pour la téléformation. Réseaux et télécommunications [cs.NI]. Université Paul Sabatier - Toulouse III, 1996. Français. NNT: . tel-00175248

HAL Id: tel-00175248

<https://theses.hal.science/tel-00175248>

Submitted on 27 Sep 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Année 1996

THESE

Présentée au

Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

en vue de l'obtention du

Doctorat de l'Université Paul Sabatier de Toulouse III (Sciences)

Spécialité : **Informatique**

par **Philippe OWEZARSKI**

Conception et formalisation d'une application de visioconférence coopérative. Application et extension pour la téléformation

Soutenue le 20 décembre 1996, devant le jury :

G.	JUANOLE	Président
M.	BANÂTRE	Rapporteur
R.	CASTANET	Rapporteur
M.	DIAZ	Directeur de thèse
C.	BETOURNÉ	Examineur
J.P.	COURTIAT	Examineur
C.	DIOT	Examineur
J.F.	SCHMIDT	Examineur
D.	WARTELLE	Invité

Rapport LAAS N° 96478

Thèse préparée au Laboratoire d'Analyse et d'Architecture des systèmes du CNRS

7, Avenue du Colonel Roche

31077 TOULOUSE Cedex 04

A Maman

Felix qui potuit rerum cognoscere causas

Virgile

**(Heureux celui qui a pu pénétrer
les causes secrètes des choses)**

Remerciements

Cette thèse a été préparée au Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique (LAAS du CNRS). Durant les 4 années que j'ai passées au LAAS (un an de DEA et 3 ans de thèse), le directeur était M. Alain Costes que je tiens à remercier très sincèrement pour son accueil.

Je tiens par dessus tout à remercier M. Michel Diaz, directeur de recherche au CNRS, de m'avoir accueilli dans le groupe OLC (Outils Logiciels pour la Communication) qu'il dirige. Au delà, de l'aide qu'il a pu m'apporter tout au long de mon séjour au LAAS, et de tout ce que j'ai pu apprendre à son contact, je tiens à le remercier pour la confiance qu'il a manifesté à mon égard en me supportant financièrement pendant 3 ans, et en me donnant au fur et à mesure de l'avancement de ma thèse de plus en plus de responsabilités dans les projets du groupe. Par ces quelques mots, je tiens à lui dire combien j'ai apprécié de travailler avec lui, et qu'il fait partie des personnes qui comptent beaucoup pour moi.

Je tiens également à remercier MM. Durieux et Vignolle, tous deux professeurs à l'université Paul sabatier de Toulouse, de m'avoir accepté au sein de la formation et de l'école doctorale qu'ils dirigent.

Je tiens également à remercier les membres du jury

- M. Michel Banâtre, directeur de recherche INRIA à l'IRISA ;
- M. Claude Betourné, professeur à l'université Paul sabatier de Toulouse ;
- M. Richard Castanet, professeur à l'université de Bordeaux I ;
- M. Jean-Pierre Courtiat, chargé de recherche au LAAS-CNRS ;
- M. Christophe Diot, chargé de recherche à l'INRIA de Sophia Antipolis ;
- M. Guy Juanole, professeur à l'université Paul Sabatier de Toulouse ;
- M. Jean-François Schmidt, ingénieur à Airbus training ;
- M. Dominique Wartelle ingénieur à Siemens Automotive

d'avoir accepté d'examiner mon travail de thèse. Je tiens à leur exprimer toute ma gratitude. En particulier, je tiens à remercier MM. Michel Banâtre, Richard Castanet et Serge Fdida d'avoir bien voulu rapporter sur ma thèse.

Je veux également remercier les personnes du LAAS avec lesquelles j'ai travaillé, et en particulier Véronique Baudin pour l'aide continue qu'elle a pu m'apporter et notamment sur l'environnement de téléformation décrit au chapitre 5 de la thèse. Je remercie aussi Michel Fournier (mon collègue de bureau et co-président du GDCQI) pour toutes les discussions "système" et "ATM" que nous avons eues avoir, et pour tous les moments passés devant des stations de travail à déboguer des programmes. Je tiens également à remercier Christophe Chassot avec lequel j'ai travaillé sur tous les aspects communications à ordre partiel ; en particulier, je tiens à le remercier de la générosité dont il a fait preuve à mon égard et du sacrifice qu'il a consenti en refusant, malgré un désir très fort, d'écrire une seule ligne du code de PNSVS 2. Je remercie également Thierry Villemur sans le concours duquel N-TSVS n'aurait jamais pu exister. Enfin, je remercie Patrick Sénac pour tout le travail dans le domaine de la synchronisation

multimédia et de la modélisation que nous avons pu faire, et pour tous les conseils qu'il a pus me dispenser.

Je remercie également tous les stagiaires qui ont travaillé avec moi tout au long de ma thèse : David Naïm, Eric Guillochin, Laurent Robert, Laurent Rocher, Jean-Christophe Arnu et Marc Boyer. Qu'ils sachent que sans eux, je n'aurais jamais pu aller aussi loin dans ma recherche.

Je remercie également tous les membres du groupe OLC pour leur accueil et pour les aides qu'ils ont pu m'apporter. En particulier, je tiens à souligner la patience de Brigitte Pradin qui partage le même bureau que moi et qui a eu à supporter mes sautes d'humeur et mes ralleries continuelles.

Je tiens également à remercier l'administration système du laboratoire pour leur gentillesse et leur efficacité. Je remercie tout particulièrement Marie-Jeanne, Danielle, Marido, Charles, Jean-Michel, Marc et Matthieu pour toutes les fois où ils m'ont dépanné.

Je tiens également à remercier la direction du LAAS, le service documentation, la gestion du personnel et la gestion financière pour leur amabilité qui n'a d'égal que leur efficacité.

Je ne peux pas non plus ne pas parler des activités extra-professionnelles que j'ai pues mener durant ma thèse. En particulier, j'ai une pensée pour toutes les VTL (Véloces Tortues du LAAS) - l'équipe de coureurs à pieds du LAAS la plus connue au monde. Je les remercie pour la confiance qu'ils m'ont accordée en me nommant "capitaine des tortues" ; je tiens à leur faire part de mon amitié même si j'en ai souvent bavé pour franchir les cols ariégeois et si j'ai des raisons de leur en vouloir (surtout Frédéric). J'ai également une pensée pour mes amis, pour toutes les soirées que nous avons passées ensemble et pour les multiples raids à pieds ou à ski que nous avons faits dans les Pyrénées.

Je tiens également à remercier mon entourage familial :

- Stéphane, mon "petit" frère, qui vient d'intégrer le groupe OLC, et auquel je souhaite de soutenir lui aussi une thèse très bientôt ;
- Anne, ma "petite-amie" qui illumine ma vie ;
- et surtout, je tiens à remercier celle à qui je dois tant, la plus formidable des mamans...

Maman, cette thèse elle est aussi à toi..

Table des matières

Introduction générale	1
------------------------------------	---

Chapitre I : Les applications distribuées multimédias coopératives : caractéristiques et besoins	5
---	---

Introduction.....	5
1. Les applications multimédias	6
1.1. Informations multimédias : caractéristiques et besoins	6
1.1.1. Notion de flux, flux continus, flux discrets	6
1.1.2. La notion de qualité de service	6
1.1.3. La synchronisation multimédia	7
1.2. Les systèmes distribués asynchrones	8
1.2.1. Pourquoi des systèmes distribués asynchrones ?	8
1.2.2. Caractérisation des systèmes asynchrones	9
1.2.3. Les problèmes engendrés par les systèmes asynchrones.....	12
1.3. Modélisation des applications multimédias	12
1.3.1. Importance de la modélisation des contraintes de synchronisation et de QoS	12
1.3.2. Les modèles existants	13
1.3.3. Le modèle TSPN	17
1.3.4. Utilisation des TSPN dans cette thèse	22
2. La coopération	25
2.1. Introduction sur la coopération et le travail coopératif	25
2.1.1. CSCW et collectif	25
2.1.2. Coopération synchrone, coopération asynchrone	27
2.2. Caractéristiques des applications coopératives	27
2.2.1. Schéma de découpage	27
2.2.2. Communications.....	28
2.2.3. Structuration des groupes	28
2.2.4. Gestion des données	29
2.2.5. Attribution des tâches	29
2.2.6. Mécanismes décisionnels	29
2.3. Modélisation des activités coopératives.....	30
2.3.1. Les modèles existants	30
2.3.1.1. La logique	30

2.3.1.2. Réseaux de Petri	30
2.3.1.3. Acteurs.....	31
2.3.2. Un modèle de coopération basé sur le partage d'informations	31
2.3.2.1. Modèle de systèmes coopératifs	31
2.3.2.2. Dépendances des données et prédicats	33
2.3.2.3. Domaines de coopération	34
2.3.2.4. Dynamicité du graphe de coopération dans le temps	35
3. Exemples d'applications multimédias.....	37
3.1. Les applications multimédias locales.....	37
3.2. Les applications multimédias distribuées	37
3.3. Le cas particulier de la visioconférence	39
Conclusion	41

Chapitre II : Gestion de la synchronisation multimédia et de la qualité de service dans une application de visioconférence

43

Introduction.....	43
1. Modélisation des mécanismes de gestion de la qualité de service	45
1.1. Mécanismes de synchronisation multimédia	45
1.1.1. Latence des cartes multimédias et décalage des synchronisations inter-flux	45
1.1.2. Calcul du décalage des rendez-vous et de la période de synchronisation du TSPN application	47
1.2. Réduction du nombre de discontinuités et contrôle du retard de présentation de bout en bout	48
1.2.1. Principe	48
1.2.2. Mécanisme de résorption des retards	50
1.2.3. Contrôle de la synchronisation inter-flux dans le mécanisme de contrôle du retard de présentation	51
1.2.4. Optimisation de la gestion des retards.....	51
1.3. Modélisation hiérarchique de l'application par des TSPN	53
2. Implémentation des mécanismes de synchronisation	54
2.1. Architecture de synchronisation.....	54
2.2. Algorithmes de synchronisation.....	57
2.2.1. Algorithme de synchronisation intra-flux	57
2.2.2. Algorithme de synchronisation inter-flux	63
2.3. Définition complète des TSPN applicatifs	65
3. Définition et négociation de la QoS dans PNSVS 1.....	66
3.1. Définition de la QoS	66
3.2. Renégociation de la QoS.....	67
3.2.1. Contraintes.....	67
3.2.2. Principe général du protocole.....	67
3.2.3. Détails du protocole de renégociation	68
3.2.4. Le décalage d'indices.....	70
3.2.5. Renégociation et contrôle du retard de bout en bout.....	71
4. Evaluation des mécanismes de synchronisation et de renégociation de la QoS de PNSVS 1	72
4.1. Mesures de la gigue et de la dérive inter-flux	72
4.2. Mesures de la gigue au dessus d'un réseau non fiable	74
4.3. Mesures des dates de présentation des objets de PNSVS 1 lors de changement de QoS	75

Conclusion	76
------------------	----

Chapitre III : Utilisation d'un transport à ordre partiel pour optimiser la qualité de service d'une application de visioconférence.....	77
---	-----------

Introduction.....	77
1. Les problèmes	78
2. Une solution basée sur un transport à ordre partiel	79
3. Architecture	81
4. Description des services fournis par le transport à ordre partiel et la pré-synchronisation	83
4.1. Services fournis par la pré-synchronisation	83
4.2. Services fournis par le transport à ordre partiel	84
5. Modélisation des niveaux de synchronisation	86
5.1. TSPN applicatif	86
5.2. TSPN de pré-synchronisation	87
5.3. Réseau de Petri des compositions série / parallèle	88
5.3. Exemple complet.....	88
6. Implémentation	89
6.1. Architecture et algorithmes de l'application de synchronisation	90
6.2. Pré-synchronisation	92
6.3. Transport à ordre partiel.....	94
Conclusion	94

Chapitre IV : Conception et implémentation d'un système de visioconférence coopératif	95
--	-----------

Introduction.....	95
1. Principes de N-TSVS.....	96
2. La gestion de la coopération dans l'application N-TSVS	98
2.1. Le protocole et le service de coopération.....	98
2.1.1. Description du service	98
2.1.2. Fonctionnement du service.....	99
2.1.3. Implantation du protocole et du service de coopération.....	101
2.2. L'interface de coopération	101
2.3. La coopération dans la partie traitement des données de N-TSVS	103
2.4. Interactions entre les modules	104
3. La partie données de la visioconférence	105
3.1. Les communications multipoints	106
3.2. La synchronisation multimédia dans N-TSVS.....	107
3.3. Le multiplexage audio.....	108
3.4. La gestion de la QoS et de la synchronisation à N.....	108
3.4.1. Gestion de la QoS à N	108
3.4.2. Gestion de la synchronisation multimédia à N.....	109
4. Traitement des erreurs dans N-TSVS	110
4.1. Les erreurs dans N-TSVS	110
4.2. Le traitement des erreurs.....	110
4.3. Détection et propagation des erreurs locales.....	112

5. Structuration de la visioconférence	112
Conclusion	113

Chapitre V : Vers la conception et le développement d'un système synchrone de téléformation professionnelle	115
--	------------

Introduction.....	115
1. Présentation de quelques exemples de systèmes d'EAO et de télé-enseignement	116
1.1. Les systèmes d'enseignement assisté par ordinateur.....	116
1.1.1. Les systèmes d'enseignement assisté par ordinateur "textuels"	116
1.1.2. Les systèmes d'enseignement assisté par ordinateur "multimédia".....	117
1.2. Les systèmes de télé-enseignement.....	117
1.2.1. Systèmes de télé-enseignement asynchrones	117
1.2.2. Les systèmes de télé-enseignement synchrones	118
2. L'application VACBI	119
2.1. L'application VACBI actuelle.....	120
2.2. Les extensions proposées pour l'application VACBI.....	121
3. Etude des interactions entre instructeur et stagiaires	122
3.1. Interactions instructeur/stagiaire(s) dans un contexte de briefing/debriefing	122
3.2. Interactions instructeur/stagiaire(s) dans un contexte d'apprentissage VACBI	124
4. Les outils.....	126
4.1. L'outil de visioconférence	126
4.2. Le tableau de dialogue	127
4.2.1. Les fonctionnalités	127
4.2.2. Les problèmes de synchronisation	128
4.2.3. Principes de fonctionnement du tableau de dialogue	128
5. Intégration et interface	129
5.2. Interfaces	130
5.2.1. Interface du poste de l'instructeur	130
5.2.2. Interface du poste d'un stagiaire	132
6. Etat des outils réalisés.....	132
Conclusion	134

Conclusion générale.....	137
---------------------------------	------------

Bibliographie de l'auteur	141
--	------------

Bibliographie	143
----------------------------	------------

Introduction générale

Ces dernières années ont connu une véritable révolution technologique avec l'explosion des systèmes *multimédias* et des *réseaux hauts débits*. A la base, ceci a été rendu possible par les nouvelles techniques de digitalisation qui ont permis de coder de façon numérique tous les médias ; on parle d'un monde de l'information "*tout numérique*". A partir de là, tous les médias, en particulier l'audio et la vidéo, ont pu être intégrés et traités par des ordinateurs. De plus, la croissance incessante de la puissance des processeurs, les nouvelles fonctionnalités des systèmes opératoires, les processeurs spécialisés capables de traiter les nouveaux types de données multimédias numérisées, les algorithmes de compression des médias les plus volumineux, les nouvelles technologies de stockage (CD-ROM) ont permis l'avènement des applications multimédias locales. La numérisation a également touché le monde des télécommunications et de plus en plus, les réseaux longues distances, eux aussi numériques, permettent le transport de tous les types de données sur le même support (par exemple le RNIS¹). Les progrès dans les techniques de communication, et en particulier la croissance de la bande passante, du débit, et l'intégration de nouveaux services ont conduit à l'explosion des réseaux hauts débits multimédias comme Internet ou ATM². On parvient donc à un nouveau monde dans lequel l'informatique et les télécommunications se rejoignent pour donner naissance aux *systèmes distribués hauts débits multimédias*.

L'utilisation conjointe des réseaux et du multimédia permet de plus d'aborder le domaine du travail coopératif assisté par ordinateur en permettant à des utilisateurs distants de travailler coopérativement sur des documents conjoints. C'est le cas par exemple dans le cadre du télé-travail, de la télé-formation, du télé-enseignement, du télé-diagnostic médical, de la télé-conception, etc.

Ces progrès dans les technologies multimédias et dans les télécommunications numériques ouvrent donc la voie à de nouvelles applications qui sont :

1. multimédias,
2. distribuées,
3. et coopératives.

Le projet CESAME

¹RNIS : Réseau Numérique à Intégration de Services.

²ATM : Asynchronous Transfer Mode.

C'est dans cet environnement que le projet CESAME³ [DIA94d] [BAU95] a débuté. Ce projet, collaboration entre le CNET⁴, le CCETT⁵ et le CNRS⁶ représenté par 9 laboratoires⁷, avait pour objectif de développer les études conceptuelles dans les systèmes distribués hauts débits multimédias coopératifs. Ce projet a pour cela été découpé en 5 tâches qui devaient s'appuyer sur le réseau ATM, ses différentes fonctionnalités et ses différents services (AAL⁸). Ces 5 tâches étaient :

- l'étude et la conception de protocoles de transport haute vitesse multimédias ;
- la définition et le développement de mécanismes de synchronisation multimédia (spatiale, temporelle, etc.) ;
- la définition et la conception de mécanismes de coopération de groupe permettant à plusieurs individus de travailler ensemble sur un même projet ;
- la définition de modèles permettant l'évaluation de performances des systèmes proposés ;
- enfin, la définition et la réalisation d'une application réelle illustrant tous les concepts étudiés dans CESAME. L'application retenue est une télé-formation professionnelle aéronautique pour la formation des pilotes et agents de maintenance sur la gamme des avions Airbus. Cette étude se poursuit dans le cadre du projet TOPASE⁹ dont le LAAS est l'un des participants.

Il faut noter toutefois que l'un des aspects essentiels qui se retrouve dans toutes ces tâches est l'aspect formel qui en découle ; tous les mécanismes ont été modélisés à l'aide d'une technique de description formelle (Estelle, réseaux de Petri, Lotos, etc.) qui permet d'exprimer sans ambiguïté les caractéristiques du système conçu, et aussi de le simuler, de le vérifier et de le valider.

Les contributions de cette thèse

Cette thèse s'inscrit dans l'ensemble des études menées, dans CESAME, sur la conception d'applications multimédias coopératives. Nous verrons qu'elle apporte donc une contribution à la tâche 5 du projet CESAME (conception d'applications), mais également aux tâches 1 (avec l'utilisation de services transport), 2 (avec la conception de mécanismes de synchronisation multimédia pour les flux de l'application) et 3 (avec le développement de mécanismes de coopération de groupe) ; en fait, par la nature même des applications multimédias coopératives, cette thèse propose la conception et l'intégration de bon nombre des résultats de CESAME dans la conception d'une application.

Les difficultés qui seront résolues dans cette thèse sont :

- l'intégration de mécanismes de gestion de la qualité de service multimédia et de la synchronisation entre les flux de l'application ; en effet, par la nature même des nouveaux médias (audio et vidéo en particulier), de nouveaux critères de qualité, et en particulier des critères temporels, devront être respectés ;
- l'intégration de mécanismes de coopération qui doivent permettre à un groupe d'utilisateurs de travailler ensemble sur un document partagé ou sur un même projet ; pour cela, après avoir étudié la façon dont les gens travaillent, des mécanismes permettant des interactions aussi riches que possibles doivent être développés ;

³CESAME : Conception formelle de Systèmes hAutS débits Multimédias coopÉratifs.

⁴CNET : Centre National d'Etude des Télécommunications.

⁵CCETT : Centre Commun d'Etude des Télécommunications et de Télédiffusion.

⁶CNRS : Centre National de la Recherche Scientifique.

⁷Les 9 laboratoires du CNRS impliqués dans CESAME sont : l'IRISA, l'IRIT, le LAAS, le LaBRI, le LIB, le LRI, le MASI, PRISM et TASC.

⁸AAL : ATM Adaptative Layer.

⁹TOPASE : Téléformation prOfessionnelle Pour l'Aéronautique et l'eSpacE.

- la conception et la réalisation d'une application de visioconférence à N participants gérés dynamiquement.

Tous les mécanismes de gestion de la qualité de service, de synchronisation multimédia et de coopération ont été appliqués à l'application de visioconférence. Ce choix s'est imposé pour deux raisons :

- tout d'abord, la visioconférence est un outil de base dans la conception de presque toutes les applications coopératives multimédias complexes (télé-formation, télé-conception, etc.), et est l'application symbole des systèmes distribués hauts débits multimédias coopératifs ;
- la visioconférence est également une application très simple, mais qui requiert de résoudre, pour être implémentée, toutes les difficultés techniques et conceptuelles liées à la manipulation de données multimédias et à la coopération. Elle permettra donc aussi de préciser les choix formels effectués, en particulier concernant des modélisations à plusieurs niveaux, et de les valider par rapport à une réalisation effective.

Aussi, la visioconférence servira de "fil rouge" dans cette thèse, et toutes les idées exposées seront appliquées à ce type d'applications. Toutefois, le dernier chapitre décrit la conception d'un système plus complexe de télé-formation professionnelle dans le milieu de l'aéronautique, lequel intègre les systèmes de visioconférence qui ont été décrits dans les chapitres précédents.

Plan de la thèse

- Tout d'abord, le premier chapitre expose la double problématique associée à nos travaux, à savoir (1) la problématique liée à la manipulation et au traitement de données multimédias qui nécessitent de concevoir des mécanismes pour gérer la qualité de service et pour garantir les contraintes de synchronisation, et (2) la problématique liée aux activités coopératives qui doit permettre aux utilisateurs de communiquer et d'interagir entre eux.

Dans les deux cas, nous nous sommes placés dans le cadre des systèmes distribués asynchrones, cadre qui constitue l'environnement le plus général et le plus vaste que l'on puisse rencontrer. Pour chacune des problématiques, un état de l'art est présenté, et un modèle est proposé. Pour la synchronisation multimédia, le modèle retenu est une extension du modèle réseau de Petri temporel : le modèle TSPN¹⁰. Pour la coopération, le modèle retenu est basé sur des graphes et sur la notion de groupes et domaines coopératifs.

Enfin, ce premier chapitre se termine par une présentation des principales applications multimédias qui existent, et en particulier par une étude de plusieurs systèmes de visioconférence, confrontés à la double problématique exposée.

- Le second chapitre présente des mécanismes qui permettent de gérer la qualité de service (comme la qualité de présentation, le délai de présentation de bout en bout, etc.) et surtout de garantir la synchronisation multimédia (intra et inter-flux). En particulier, le TSPN qui permet d'exprimer les contraintes de synchronisation a été dérivé pour donner naissance aux TSPNs applicatifs qui modélisent le comportement des processus de l'application exécutant les mécanismes de synchronisation multimédia. Ces mécanismes ont été développés en utilisant les techniques avancées des systèmes opératoires normalisés Posix, comme les "threads", les classes d'ordonnancement de processus temps réel, etc. et ont conduit à l'élaboration du logiciel de visioconférence PNSVS¹¹ au dessus du protocole transport UDP¹². De plus, la qualité de service est dynamique et pourra être renégociée à tout moment, suite à une requête des utilisateurs

¹⁰TSPN : Time Streams Petri Net.

¹¹PNSVS : Petri Net Synchronized Videoconference System.

¹²UDP : User Datagram Protocol [POS80].

ou suite à une modification des ressources disponibles (au niveau du système ou du réseau). Ces mécanismes sont décrits dans ce chapitre.

- Le chapitre 3 montre les insuffisances des mécanismes de synchronisation de PNSVS 1, qui, bien qu'ils implémentent de façon stricte les procédés permettant de garantir les contraintes de synchronisation, engendrent parfois de fortes dégradations de la qualité de service. Les causes de cette dégradation sont dues au transport utilisé (UDP) qui s'avère inadapté. Dès lors, les mécanismes proposés au chapitre 2 sont enrichis de façon à éviter ces dégradations intolérables ; pour cela, le service transport UDP est remplacé par un transport à ordre partiel qui grâce à des mécanismes de fiabilité partielle et de perte et délivrance des objets au plus tôt permet d'éviter ces dégradations et conduit à la version 2 de PNSVS.
- Le chapitre 4 ajoute des notions de coopération à la visioconférence. Pour cela, des mécanismes de gestion de la coopération sont introduits, comme des mécanismes d'entrée ou de sortie dynamique d'un groupe coopératif. De même, des mécanismes de contrôle des interactions, de communication de groupe et de gestion de la qualité de service à N sont présentés. Ces mécanismes ont conduit à l'élaboration du système de visioconférence N-TSVS¹³.
- Enfin, le cinquième et dernier chapitre intègre la visioconférence dans un système applicatif plus large. Il présente la conception d'une application de télé-formation synchrone dans le domaine de l'aéronautique. L'objectif est d'assurer au travers de stations de travail et de réseaux de communication la future formation des pilotes et agents de maintenance sur la gamme des avions Airbus. Pour cela, des outils de coopération ont été définis ; ces outils sont la visioconférence point à point ou multipoints et un tableau de dialogue électronique qui remplace le traditionnel tableau noir des salles de cours. De plus, les interfaces des postes élèves et instructeurs sont définies et l'état d'avancement de cet environnement est décrit. La finalisation et l'expérimentation de ce système font l'objet du projet TOPASE précédemment introduit.

¹³TSVS : Timestamp Synchronized Visioconference System.

Chapitre I

Les applications distribuées multimédias coopératives : caractéristiques et besoins

Introduction

Ces dernières années, le monde de l'informatique a été marqué par l'avènement "*du multimédia*" et des "*autoroutes de l'information*" qui permettent les traitements et les communications à hauts débits des nouvelles informations multimédias numériques. Les progrès dans ces deux domaines ont conjointement contribué à l'essor d'un nouveau type d'applications : *les applications distribuées multimédias coopératives* qui se caractérisent par :

- l'utilisation conjointe et massive de plusieurs types de données comme le texte, les graphiques, mais aussi des données plus nouvelles comme les images fixes, et surtout l'audio et la vidéo ;
- L'utilisation des nouveaux supports de communication qui permettent à des utilisateurs de communiquer grâce à des outils de plus en plus riches, interactifs, conviviaux et efficaces.

Toutefois, les aspects multimédias et coopératifs de ces applications posent un grand nombre de problèmes qu'il faudra résoudre. Aussi, l'objectif de ce chapitre est de dégager la problématique inhérente à la conception d'applications distribuées multimédias coopératives.

Ce chapitre est structuré en trois parties.

Tout d'abord, comme introduit ci-dessus, la problématique est double puisque les applications qui vont faire l'objet de cette thèse intègrent à la fois des aspects multimédias et des aspects coopératifs. Aussi, dans la première partie, les caractéristiques et exigences du multimédia seront posées. La seconde partie exposera la problématique associée à la coopération et aux activités coopératives. Enfin, la troisième partie présentera quelques applications multimédias coopératives, puis elle s'attardera un peu sur le cas particulier de la visioconférence, la visioconférence étant l'application que nous avons choisie pour illustrer les concepts qui seront présentés dans les prochains chapitres.

1. Les applications multimédias

1.1. Informations multimédias : caractéristiques et besoins

1.1.1. Notion de flux, flux continus, flux discrets

La première caractéristique qui distingue les données multimédias des données informatiques classiques (textes, données binaires, ...) est leur unité de traitement : les données multimédias se manipulent par *flux*, alors que les textes et les données binaires se manipulent par fichiers. En effet, les données audio et vidéo sont des séquences d'images ou d'échantillons sonores qui se succèdent à une cadence constante ou non. Ces données ne sont pas incompatibles avec la notion de fichier informatique, et les fichiers sont toujours utilisés pour stocker des films ou des documents sonores ; mais les traitements se font unité d'information par unité d'information (image par image, par exemple) ; ces unités d'information mises bout à bout forment un flux. Il faut noter toutefois que les textes, graphiques et données binaires s'accommodent de ce mode de traitement.

Cependant, les flux se caractérisent aussi par les relations temporelles qui existent entre les différentes unités d'information qui les composent. Par exemple, il n'existe pas de relation temporelle entre les caractères qui composent un flux textuel. De même, pour une image unique qui peut être vue comme un flux de bits, ou pour un graphique, il n'existe aucune relation temporelle entre les différentes unités du flux. Ce sont typiquement des *flux discrets*.

Par contre, pour la vidéo ou le son, les images ou les échantillons sonores doivent être produits, traités ou présentés avec une cadence régulière. On parle dans ce cas là de *flux continus*. Si le temps entre deux unités de base d'un flux est constant, on parle de flux *isochrone* ; toutefois, une certaine variabilité sur ces temps peut être tolérée : cette variabilité est appelée *gigue autorisée*.

1.1.2. La notion de qualité de service

Dans la présentation de la notion de flux qui vient d'être faite sont déjà apparues des notions de contraintes temporelles. Ces contraintes ne sont pas les seules et d'autres paramètres de qualité existent. En fait, les flux multimédias se caractérisent par leur qualité de service (ou QoS¹ en anglais). Une des familles de paramètres de qualité de service concerne la qualité de restitution des flux multimédias.

Par exemple, les données binaires ou textuelles ne tolèrent aucune perte (elles nécessitent une fiabilité totale) ; par contre, leurs exigences en terme de capacité de stockage ou le débit requis pour les transmettre sur un réseau sont très faibles, et ce type de données ne présente aucune contrainte temporelle.

De manière identique, les graphiques supportent très mal les pertes ou les erreurs. Ils ne requièrent que peu de ressources de stockage ou de communication, et sont relativement insensibles aux phénomènes de gigue.

Les images fixes, quant à elles, tolèrent assez bien les erreurs ou la perte de certaines informations. Toutefois, la capacité de stockage ou le débit requis sur un réseau de communication sont supérieurs à ce qui est nécessaire pour du texte ou du graphique. D'ailleurs pour réduire la quantité de données d'une image vidéo, des algorithmes de compression ont été conçus comme GIF ou JPEG² [WAL91].

Le son, lui, supporte difficilement les erreurs ou les pertes ; ces parasites nuisent de façon relativement importante à la qualité du flux audio. Le débit requis et la capacité

¹QoS : Quality of Service.

²JPEG : Joint Photographic Expert Group.

de stockage sont variables en fonction du codage utilisé : seulement 64 kbps sont nécessaires pour du son de la qualité du téléphone numérique, alors que 1,2 Mbps (sans compression) sont requis pour du son Hi-fi stéréophonique de qualité laser. De plus, le flux audio est très sensible aux perturbations temporelles qu'il peut subir.

Enfin, la vidéo animée est le média le moins contraignant en terme d'erreur et de perte : la perte d'une image n'est pas perceptible par l'utilisateur terminal. Par contre la quantité de données engendrée et le débit requis sont considérables. D'ailleurs des efforts importants ont été effectués dans le secteur de la compression vidéo avec les algorithmes H261 [LIO91] [TUR93], MPEG³ [LEG91], etc. ; toutefois, les flux vidéo, même compressés, restent extrêmement gourmands en place mémoire et en bande passante. De plus, les contraintes temporelles sur les images d'un flux vidéo sont assez fortes, même si un certain niveau de gigue reste acceptable.

Ce qui précède vient de faire un petit inventaire des contraintes de qualités de service des différents médias. Toutefois, cet inventaire n'est pas exhaustif et ne prend en compte que les paramètres de QoS en rapport avec les pertes, les erreurs, la place mémoire et la bande passante requise. De toute manière, énumérer tous les paramètres de qualité de service est impossible car chaque application aura ses propres besoins, et aura donc ses propres paramètres de QoS à respecter sur ses données propres. Nous nous contentons donc de présenter les paramètres de QoS les plus habituels dans les applications multimédias actuelles comme la visioconférence.

1.1.3. La synchronisation multimédia

La partie précédente vient de mettre en évidence la notion de QoS multimédia et l'a illustrée en présentant les principaux paramètres de qualité de présentation associés aux flux multimédias. Toutefois, l'aspect temporel qui a été signalé dans la partie 1.1.1 est essentiel, et les flux continus se caractérisent surtout par leurs contraintes temporelles. Par exemple, dans une présentation multimédia qui comporte plusieurs types de données, il existe des relations temporelles et spatiales dans les différents flux. Il peut également exister des relations temporelles et spatiales entre ces mêmes flux. Ces relations définissent en fait les contraintes de synchronisation multimédia. La suite va en présenter les différents aspects.

La synchronisation spatiale

La synchronisation spatiale exprime les contraintes d'ordonnement visuel des différents objets multimédias sur le support de présentation (écran, mur d'image, etc.). Elle permet donc de définir la taille des différentes zones, les superpositions, les juxtapositions, etc. comme dans la norme MHEG⁴ [ISO90] [ISO93].

La synchronisation temporelle

La résolution des problèmes liés à la synchronisation temporelle constitue le point essentiel et le plus délicat dans la conception des systèmes multimédias [BLA96]. Il s'agit d'exprimer et de garantir les contraintes et relations temporelles qui existent entre les objets d'un document multimédia. Deux types de synchronisations apparaissent : la synchronisation intra-flux et la synchronisation inter-flux.

La synchronisation intra-flux consiste à respecter les contraintes de présentation sur chacune des unités d'information d'un flux ; ceci revient donc à contrôler la gigue de sorte que le temps de présentation ne s'écarte pas de sa valeur idéale d'une valeur supérieure à la gigue maximale autorisée. Par exemple, pour une animation vidéo, il faut garantir que le temps de présentation de chaque objet vidéo (image) respectera le temps qui existait entre deux captures lors de la création de l'animation.

³MPEG : Motion Pictures Expert Group.

⁴MHEG : Multimedia and Hypermedia information coding Expert Group.

La synchronisation inter-flux consiste à contrôler la dérive (ou décalage temporel) qui peut exister entre deux flux. La dérive est due à l'effet cumulatif de la gigue ; en effet, les giges que subissent chacun des objets du flux peuvent s'accumuler, et le décalage qui peut apparaître entre deux flux peut devenir très grand. Il faut donc contrôler cette dérive et faire en sorte qu'elle reste en deçà d'un seuil de tolérance. C'est typiquement le cas de la synchronisation entre un flux audio et un flux vidéo, pour lesquels il faut assurer que les sons correspondent bien aux mouvements des lèvres ; ce problème est connu sous le nom de "synchronisation des lèvres".

De même, il existe une séparation entre la synchronisation discrète et la synchronisation continue.

La synchronisation discrète apparaît la plupart du temps en présence d'un flux discret ; elle consiste à synchroniser des objets lorsque le besoin s'en fait sentir. Par exemple, dans le cadre d'un film sous-titré, il faut synchroniser le texte du sous-titrage avec le film, ce qui n'est nécessaire que lorsqu'il y a des dialogues.

En revanche, la synchronisation continue consiste à introduire périodiquement des points de synchronisation dans la présentation du ou des flux. Par exemple, dans le cas de la synchronisation des lèvres, il faut périodiquement introduire des points de synchronisation entre les flux audio et vidéo pour éviter que la dérive ne croisse trop.

La synchronisation hypermédia

La synchronisation hypermédia intègre les notions de synchronisation spatiale et de synchronisation temporelle. Toutefois, elle ajoute à ces notions la notion de synchronisation logique qui consiste à synchroniser les présentations de l'application lors du parcours des liens d'un document hypermédia (comme dans MHEG [ISO90] [ISO93]). Elle ne sera toutefois pas considérée dans cette thèse.

1.2. Les systèmes distribués asynchrones

1.2.1. Pourquoi des systèmes distribués asynchrones ?

L'étude et les travaux de synchronisation décrits dans cette thèse ont été effectués dans le cadre des systèmes distribués asynchrones. Ce choix s'est imposé pour trois raisons :

- Tout d'abord, les systèmes distribués asynchrones représentent à quelques exceptions près la totalité des systèmes opérationnels existant à l'échelle planétaire. Les réseaux locaux Ethernet, FDDI, Token Ring, ... ainsi que les lignes Internet qui les interconnectent sont asynchrones ; les systèmes opératoires qui s'exécutent sur les stations de travail sont également asynchrones (Unix, DOS, ...). De plus, la tendance actuelle (imposée par les constructeurs) ne fait qu'accroître l'asynchronisme des systèmes pour gagner en performance, et l'asynchronisme semble donc être l'avenir dans le domaine des machines de bureau : l'asynchronisme est le point de passage obligé vers les hauts débits. Aussi, nous avons souhaité faire une étude dans un cadre le plus général possible pour donner des mécanismes de synchronisation qui pourront être généralisés dans un maximum d'applications sur un maximum de sites présents et futurs ;
- Ensuite, la solution au problème de la synchronisation multimédia dans un environnement synchrone est un cas particulier de celle en environnement asynchrone (c'est le cas où l'asynchronisme est nul), et nous apportons donc une solution au problème le plus général ;
- Enfin, avec l'intégration dans UNIX (POSIX) de classes d'ordonnancement temps réel (comme dans Solaris 2), il est possible de réaliser des applications temps réel, ce qui n'était pas le cas au début de la décennie lorsque les travaux décrits dans [COU94a][JEF94b] et [JEF92] ont débuté, et avaient alors déclaré UNIX inapte pour

supporter des applications ayant des contraintes temps réel [VOG95]. Ces travaux préconisaient donc d'utiliser des systèmes temps réel qui étaient les seuls susceptibles de respecter les contraintes temporelles des applications. Il faut noter ici que la théorie sur les systèmes temps réel est une chose, mais que la réalité en est une autre [KAT94]. Dès que l'on essaie d'implémenter un système opératoire temps réel, quasiment les mêmes problèmes d'asynchronisme qu'avec un système classique (comme Solaris 2) se posent :

- les entrées / sorties sont asynchrones ;
- il peut y avoir des problèmes d'inversion de priorités lors de la synchronisation de processus ;
- le noyau doit être entièrement préemptible ;
- les tâches systèmes indispensables peuvent perturber l'ordonnancement temporel des applications ;
- etc. ...

En fait, pour obtenir un comportement temps réel avec un système temps réel, il faut se limiter à des processus de calcul pur qui ne font pas d'entrées / sorties [BAK94]. Cependant, ceci est incompatible avec les applications distribuées multimédias qui font des accès très nombreux aux cartes audio et vidéo et aux cartes réseaux. Les conditions d'utilisation d'un système temps réel sont donc trop restrictives pour des applications distribuées multimédias. En fait, utiliser des systèmes classiques (UNIX ou POSIX) revêt beaucoup d'intérêt, par leur nombre et leur disponibilité, et ne pose pas plus de problèmes que lors de l'utilisation d'un système temps réel [ADE94].

De plus, le choix des systèmes asynchrones s'est imposé car [KAN94] a montré que les systèmes synchrones à grande échelle sont impossibles à mettre en oeuvre. Ainsi, pour l'exemple de la visioconférence, qui nous intéresse tout particulièrement dans cette thèse, et qui est une application distribuée, et potentiellement longue distance, il faut donc passer par des systèmes asynchrones.

1.2.2. Caractérisation des systèmes asynchrones

L'utilisation d'un système asynchrone, qui s'avère donc être le passage obligé pour le développement des applications multimédias coopératives et en particulier celui d'une application de visioconférence, introduit néanmoins de nombreux problèmes pour traiter des données multimédias isochrones (audio et vidéo). Le problème essentiel réside dans la variabilité et l'imprévisibilité des temps de traitement des différentes opérations, et surtout ce temps de traitement est non borné supérieurement. Cette variabilité apparaît à plusieurs niveaux dans les systèmes distribués multimédias asynchrones : les trois niveaux qu'il est possible de séparer sont :

- les supports et protocoles de communication. Par exemple, les modes d'accès à des réseaux comme Ethernet, Token ring ou FDDI sont non déterministes (dépendant de la charge du réseau par exemple), et aucun délai de transit maximum n'est garanti. Ce phénomène d'asynchronisme et d'imprévisibilité est encore plus marqué lorsqu'un réseau public longue distance comme le réseau Internet est considéré ;
- le système opératoire temps partagé classique (par exemple Unix) dont la variabilité dans les temps de traitement est due :
 - aux mécanismes d'ordonnancement temps partagé du système qui tendent à privilégier les processus interactifs et le débit moyen du système d'exploitation, au détriment de certaines contraintes temps réel des processus. De plus, la notion de processus lourd telle qu'elle est utilisée dans Unix induit une surcharge en matière d'ordonnancement incompatible avec le haut degré de parallélisme et les hauts débits des données associées aux flux multimédias ;
 - au caractère non préemptif ou localement préemptif du noyau qui induit un temps de latence non déterministe pour le traitement des interruptions ;

- à la commutation des pages mémoires et de nombreuses autres tâches système qui s'exécutent à des moments non déterministes et avec des priorités supérieures aux tâches utilisateur, et qui en cela gênent la qualité temporelle de ces dernières ;
- le matériel, et en particulier les cartes multimédias (audio et vidéo). En effet, la nouvelle génération de microprocesseurs intègre des caches d'instructions et de données multi-niveaux qui génèrent des temps d'accès mémoire et de commutation de contextes non déterministes. De même, la gestion de la mémoire virtuelle introduit une variabilité des temps d'accès mémoire. Cette variabilité est renforcée dans les architectures multiprocesseurs par les mécanismes de réactualisation des caches et les contentions pour l'accès au bus. De plus, les systèmes de gestion d'interruptions hiérarchisés induisent des temps de latence non déterministes. Les disques, les processeurs spécialisés de compression / décompression et les périphériques multimédias sont ainsi autant de sources d'asynchronisme.

cartes multimédias	●	●		désynchronisations car le noyau est asynchrone
X Window		●	●	
application avec synchronisation		● pour certains appels systèmes non liés aux communications	●	Le système utilisateur est ordonnancé suivant un mode temps partagé source d'asynchronisme
UDP		●		Best Effort
IP		●		le routage IP est asynchrone (dépendant de la charge des routeurs)
liaison de donnée	LLC		●	
	MAC	●		les modes d'accès sont asynchrones
pile OSI / pile système	cartes	noyau	Espace utilisateur du système	Remarques

Figure I.1. Les sources de désynchronisation dans une application distribuée multimédia

Les systèmes distribués asynchrones introduisent donc trois grands niveaux d'asynchronisme et donc de désynchronisation. Cependant, dans une application multimédia distribuée comme la visioconférence, qui utilise une pile de protocoles de communication, il est plus habituel de diviser les différentes tâches en couches, sans se soucier de la partie du système qui réalise effectivement les opérations correspondantes. Toutefois, dans ce chapitre qui traite des techniques de gestion de la QoS et de la synchronisation de flux multimédias, il est essentiel de bien corrélérer les deux approches

(modèle du système distribué asynchrone et modèle en couche OSI) pour déterminer où sont créées les désynchronisations. La figure I.1 représente, pour le cas concret évoqué dans l'introduction (i.e. une application de visioconférence au dessus d'UDP), le lien entre les couches du système et les couches de l'application.

- Sur la figure I.1, seules les couches matérielles, noyau et système utilisateur des systèmes opératoires apparaissent. La couche physique des réseaux de communication n'apparaît donc pas. Toutefois, cette couche n'apporte, à notre échelle, aucun problème d'asynchronisme (si les phénomènes de perturbations physiques et de pertes sont négligés). Les temps de transit de l'information peuvent être connus et sont invariables. Ils dépendent simplement des vitesses d'émission et de réception, de la distance et de la vitesse de propagation des ondes (électrique, électromagnétiques, lumineuses, ...) sur leur support (câble, atmosphère, fibre optique, ...).
- Pour la sous couche MAC, par contre, suivant la méthode d'accès au médium utilisée, le comportement sera synchrone ou asynchrone.
 - Le mode d'accès CSMA/CD du réseau Ethernet est purement asynchrone. En effet, avant d'émettre, il faut attendre que le câble ne soit pas occupé, ce qui est fonction de la charge réseau et donc non déterministe. De plus, en cas de collision, la durée avant une réémission est déterminée de façon aléatoire, ce qui rend le mécanisme encore plus indéterministe ;
 - Dans le cas d'un anneau à jeton du type Token Ring, le comportement est également asynchrone. En effet, pour garantir un temps de rotation assez rapide sur l'anneau, une station ne peut garder le jeton plus de 10 ms. Si cette machine doit émettre de nombreux messages qui nécessitent un temps d'émission supérieur à 10 ms, elle devra relâcher le jeton au bout de 10 ms, et n'émettra la suite de ces messages que lorsqu'elle aura de nouveau le jeton. Ainsi, lorsqu'un utilisateur émet un paquet, le temps d'accès au médium est dépendant des quantités de données qui ont été passées à la couche MAC pour émission, et qu'il lui reste à émettre. Un mécanisme de réservation basé sur des priorités existe également dans Token Ring. Cependant, ce mécanisme n'apporte pas de solution à l'asynchronisme car le nombre de tours d'anneau que fera le jeton avant de donner le droit d'émettre à la station résérvatrice est non déterministe (fonction des réservations qui ont été faites précédemment et des réservations de priorités supérieures qui seront faites alors que la station étudiée attend son droit à émettre) ;
 - Le réseau FDDI dispose lui de deux modes d'accès : un mode asynchrone et un mode synchrone. Le mode asynchrone de FDDI ressemble au mode d'accès de Token Ring (sans réservation), et est asynchrone pour les mêmes raisons : le temps pendant lequel une station peut conserver le jeton est limité, et il n'est pas forcément suffisant pour émettre tout le trafic. Le mode synchrone est basé sur une réservation statique sur tout le réseau. Ainsi, une part des ressources du réseau est réservée au trafic synchrone (prioritaire sur le trafic asynchrone), et cette part correspond aux quantités d'informations que peut transmettre une station pendant le laps de temps pendant lequel elle détient le jeton. Cette approche possède donc l'inconvénient de limiter de façon considérable la quantité d'informations pouvant utiliser le mode synchrone de FDDI. De plus, dans les implantations du réseau FDDI qui ont été faites, le mode synchrone n'a jamais été implémenté. Les réseaux FDDI qu'il est donc possible d'utiliser aujourd'hui sont asynchrones ;
 - Le réseau DQDB, comme FDDI, normalise un trafic synchrone et un trafic asynchrone. Cependant, comme dans le cas de FDDI, le trafic synchrone n'est pas mis en oeuvre dans les implémentations actuelles. Les réseaux DQDB qu'il est possible de rencontrer sont asynchrones.
- Les couches LLC et UDP sont "best effort". En elles même, elles ne sont pas sources de désynchronisation, mais elle ne font rien pour garantir un quelconque service synchrone. Cependant, comme les processus réalisant les opérations associées à ces

couches sont ordonnancés au niveau du noyau du système opératoire (qui est asynchrone), ces couches induisent un asynchronisme.

- La couche IP est également asynchrone. En effet, au niveau des routeurs, le routage dépend de la charge du réseau, ce qui induit donc une imprévisibilité quant à la date de traitement effective des paquets reçus par le commutateur. De plus, les tâches de la couche IP sont réalisées au niveau du noyau du système opératoire, ce qui introduit donc un asynchronisme supplémentaire.
- Les tâches de la couche de synchronisation de l'application, ainsi que celles de la couche X Window sont traitées au niveau de la partie utilisateur du système opératoire, et donc dans un mode temps partagé, ce qui induit un asynchronisme considérable. De plus, ces deux couches font parfois des appels systèmes (autres que les appels système liés aux communications), traités par le noyau qui est également asynchrone.
- Enfin, les cartes multimédias induisent aussi un asynchronisme car pour y accéder, il faut traverser le noyau. D'autre part, les opérations réalisées par ces cartes étant de plus en plus complexes (par exemple, la compression vidéo), elles intègrent des processeurs et des architectures de plus en plus complexes qui introduisent donc des phénomènes d'asynchronisme (voir plus haut).

1.2.3. Les problèmes engendrés par les systèmes asynchrones

Les problèmes qui apparaissent dans les systèmes asynchrones sont donc :

- les problèmes de gigue dus à la variabilité temporelle des opérations effectuées dans le système. Ce dernier ne peut pas garantir un temps de traitement constant et/ou prévisible pour les différentes opérations et ne peut donc pas garantir les temps de présentation des données multimédias ;
- les problèmes de dérive qui sont dus à l'effet cumulatif de la gigue et peuvent faire apparaître un décalage important après le traitement d'une longue séquence d'objets d'un flux ;
- et enfin, les problèmes de pertes et de duplications ; en effet, l'ordonnancement des tâches étant imprévisible, il est possible que dans le cadre de la gestion d'un tampon, par exemple, les tâches de production accèdent plus facilement au processeur que les tâches de consommation. Ceci conduit à une saturation du tampon et à des pertes par écrasement. De même, des pertes peuvent apparaître dans les réseaux de communication. Des duplications d'objets peuvent également se produire, en particulier dans les réseaux, lorsque l'algorithme de routage duplique un paquet en plusieurs exemplaires.

Ce sont tous ces problèmes qui devront être résolus, ou du moins contrôlés, pour permettre l'écriture d'applications distribuées multimédias.

1.3. Modélisation des applications multimédias

1.3.1. Importance de la modélisation des contraintes de synchronisation et de QoS

Ce qui précède a montré l'extrême variété des contraintes de QoS qui peuvent exister. De plus, les contraintes temporelles sont relativement complexes à exprimer, et l'un des premiers problèmes à résoudre consiste à pouvoir représenter de façon simple et complète les contraintes de synchronisation qui peuvent exister dans un document multimédia.

Ce point n'était pas apparu essentiel de prime abord, et nous avons pensé pouvoir garantir la synchronisation intra et inter-flux dans un système de visioconférence sans modéliser les contraintes que nous devons respecter. Pour cela, une solution intuitive pour la synchronisation multimédia consiste à utiliser des estampilles [DIA94a] qui

contiennent les dates de présentation de chaque objet. Ainsi, chaque objet est estampillé à l'aide d'une date relative (la date 0 correspondant au lancement de l'application), et le processus de présentation n'a plus qu'à présenter l'objet considéré à la date indiquée par son estampille. Cette technique a été utilisée dans le développement d'un premier prototype d'application de visioconférence TSVS (Timestamp Synchronized Visioconference System)⁵, dont les qualités et défauts sont évalués plus avant (cette technique a par la suite été utilisée dans [ROT95]).

La technique de synchronisation par estampille est simple à implémenter, et elle résout à la fois les problèmes de synchronisation intra et inter-flux. La synchronisation intra-flux est forcément garantie, car une séquence capturée en N secondes sera jouée en N secondes avec une cadence de présentation des objets régulière. D'autre part, la synchronisation inter-flux est également obtenue, car chaque flux se synchronise par rapport à une base temporelle commune, ce qui synchronise donc les deux flux l'un par rapport à l'autre [DIA94a].

Cependant, les estampilles ne prennent pas en compte la notion d'asynchronisme. Elles n'intègrent pas la notion d'intervalle temporel, et ne peuvent pas supporter une gigue sur un objet. Ainsi, si un objet arrive quelques instants après sa date de présentation, il ne sera pas présenté, ce qui crée une discontinuité dans le flux, alors que le retard correspondait peut être à une gigue acceptable ; de ce fait, la donnée a été écartée alors qu'elle aurait pu être présentée avec un léger retard⁶. La dégradation est donc bien supérieure à ce qu'elle aurait pu être, ce qui est inadmissible puisque l'objectif est d'obtenir la meilleure qualité de présentation possible. De plus, à cause de l'asynchronisme, il est impossible de garantir des dates fixes au niveau du système opératoire. Enfin, la technique de synchronisation par estampille ne permet qu'une connaissance a posteriori des contraintes de synchronisation à respecter ; l'entité de synchronisation terminale ne connaît la date de présentation d'un objet que lorsqu'elle reçoit cet objet, et elle ne peut donc pas anticiper les traitements à mettre en oeuvre.

Ces constatations ont tout à fait conforté notre opinion : il s'avère indispensable de modéliser de façon très précise les contraintes de synchronisation multimédia que nous voulions voir respectées dans nos applications multimédias. Pour cela, nous avons été amenés à faire une étude des modèles existants pour voir s'il n'en existait pas un qui pourrait convenir à la problématique que nous nous sommes posée.

1.3.2. Les modèles existants

Pour pouvoir réaliser des applications multimédias synchronisées avec les hypothèses minimalistes que nous nous sommes fixées, il faut tenir compte de tous les problèmes liés à la variabilité des temps de traitements des systèmes asynchrones (gigue, dérive) et les relier aux propriétés intrinsèques de chaque objet multimédia (par exemple relier la gigue d'un système distribué avec la gigue admissible propre à chaque objet multimédia). Aussi, pour représenter ces propriétés de synchronisation des objets multimédias, il est essentiel de disposer d'un modèle permettant à l'auteur d'une application multimédia de spécifier les contraintes de synchronisation devant être satisfaites par l'application. Cette approche formelle est particulièrement intéressante car elle permet d'une part de spécifier sans ambiguïté des scénarios de présentations

⁵TSVS est un système de visioconférence synchronisé à l'aide de la technique par estampille. La qualité de synchronisation est très bonne et l'outil a un fonctionnement tout à fait acceptable lorsque les problèmes liés à l'asynchronisme sont réduits (i.e. les machines et le réseau ne sont pas trop chargés). Cet outil est disponible sur simple demande auprès de ses auteurs ou de la société Parallax qui assure la distribution du logiciel.

⁶Dans TSVS, les problèmes de giges sont résolus par un artifice. En effet, pour éviter d'avoir des problèmes de giges positives (qui ne peuvent être résolus), les données reçues sont stockées pendant un temps supérieur à la gigue maximale généralement observée. Cependant, ce stockage est très dommageable pour l'interactivité, car il induit un retard conséquent entre l'émetteur et le récepteur. C'est pourquoi, TSVS ne fonctionne correctement que lorsque les problèmes liés à l'asynchronisme sont réduits.

multimédias, mais elle permet également de les simuler et de les valider. De nombreuses études ont déjà été réalisées dans ce domaine, et des modèles ont été proposés [DIA93a]. En particulier, certains de ces modèles utilisent des approches formelles basées sur les réseaux de Petri temporisés dont le caractère graphique permet de mettre facilement en oeuvre des paradigmes multimédias tels que le paradigme de régie numérique [SEN94]. La partie suivante fait un panorama des extensions temporelles des réseaux de Petri en les confrontant à la problématique de la synchronisation en terme de pouvoir d'expression et de modélisation⁷. Par la suite, d'autres modèles non basés sur les réseaux de Petri seront abordés.

Modèles basés sur les réseaux de Petri

- Le modèle OCPN (Object Composition Petri Net) [LIT90] utilise des réseaux de Petri avec des places temporisées représentant les contraintes de synchronisation de base des objets multimédias. Le modèle OCPN repose sur un ensemble de 7 schémas de synchronisation de base qui caractérisent toutes les positions relatives entre deux intervalles temporels [ALL83]. Ces 7 schémas définissent l'ensemble des règles de composition utilisables pour construire des modèles complexes à partir de modèles plus simples.

Cependant, comme les OCPN utilisent des durées nominales pour les objets multimédias, il fait à la fois abstraction :

- de la variabilité temporelle induite par les systèmes asynchrones, et
- de la gigue admissible qui est intrinsèque à chaque objet. En effet, la présentation des objets d'un flux multimédia accepte une variabilité temporelle autour de sa durée de présentation nominale (i.e. la sémantique de l'unité d'information est valide dans cet intervalle).

Aussi, le modèle OCPN représente un modèle idéal qui considère toutes les durées parfaites. Il n'utilise que des durées nominales ce qui ne permet pas de modéliser la variabilité temporelle des traitements dans un système asynchrone, et limite donc son pouvoir d'expression.

- Le modèle Time Petri Net (TPN) de Merlin [BER91] rend possible l'utilisation de temporisations imparfaites en labelant chaque transition à l'aide d'un intervalle de temps dans lequel doit se situer le tir de la transition. Les règles de tir d'un TPN précisent que si une transition t , avec l'intervalle de tir $[\alpha, \beta]$, est sensibilisée à la date absolue τ , alors :

- t ne sera pas tirée avant $\tau + \alpha$
- t sera tirée au plus tard à $\tau + \beta$

Ce modèle a un très bon pouvoir d'expression, équivalent à celui d'une machine de Turing [MER74]. Cependant, il induit une synchronisation forte entre les flux qui est toujours conduite par le flux le plus en retard (la transition n'étant sensibilisée que lorsque toutes ses places en entrée sont marquées), ce qui peut entraîner le non respect des contraintes temporelles des flux les plus en avance. La figure I.2 illustre ce problème : sur cet exemple qui représente une synchronisation entre deux flux audio et vidéo, si l'on analyse le pire des cas, i.e. un scénario dans lequel on choisit le délai maximal pour tous les objets d'un flux et le délai minimal pour tous les objets de l'autre flux, il est impossible de trouver un intervalle $[x, y]$ pour la transition T qui satisfasse les sémantiques temporelles des unités d'information symbolisées par a_6 et v_3 .

⁷Le pouvoir de modélisation d'un modèle est sa capacité à représenter de façon aisée un scénario. Son pouvoir d'expression est sa capacité à spécifier le scénario de façon complète.

Ainsi, le modèle TPN ne permet pas une modélisation aisée des schémas de synchronisation inter-flux par simple fusion de transitions, et n'est donc pas adapté à la spécification de scénarios de synchronisation multimédia.

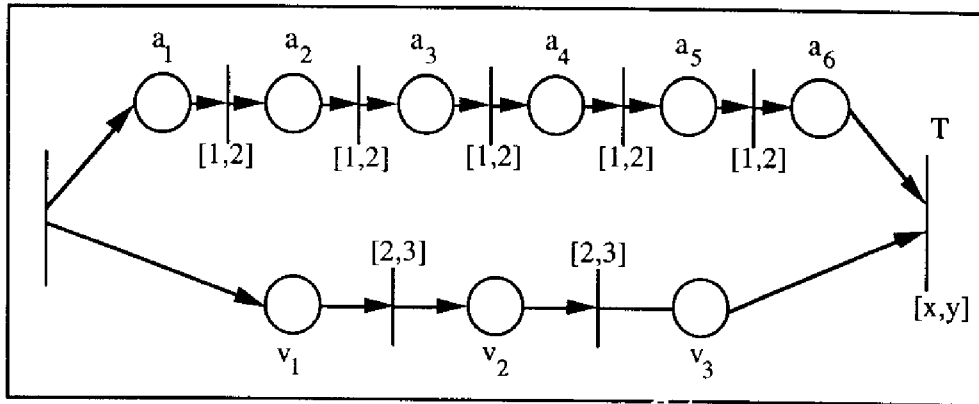


Figure I.2. Exemple de TPN

• Le modèle Arc Time Petri Net (ATPN) [WAL83] place les intervalles temporels sur les arcs sortant des places. Lorsqu'une place reçoit un jeton, une horloge locale démarre. Si une place p_i est marquée à la date absolue τ_{ij} , (p_i, t_j) étant l'arc considéré, alors si $(\alpha_{ij}, \beta_{ij})$ est l'intervalle associé à l'arc (p_i, t_j) :

- t_j ne sera pas tirée avant $\tau_{ij} + \alpha_{ij}$
- t_j sera tirée au plus tard à $\tau_{ij} + \beta_{ij}$

La différence entre les ATPN et les TPN apparaît lorsque plusieurs places sont synchronisées par une transition. Dans ce cas là, les règles de tir doivent être vérifiées pour chaque arc. La transition t_j est tirable si l'intersection de tous les intervalles dynamiques associés aux arcs n'est pas vide, i.e. s'il existe au moins une valeur du temps qui permette le tir de tous les arcs entrant de la transition t_j . Les incohérences temporelles peuvent apparaître dans [WAL83], mais aucune sémantique temporelle et aucune procédure de validation n'est donnée pour permettre de détecter ces comportements inconsistants. La figure I.3 montre un exemple d'une telle incohérence temporelle :

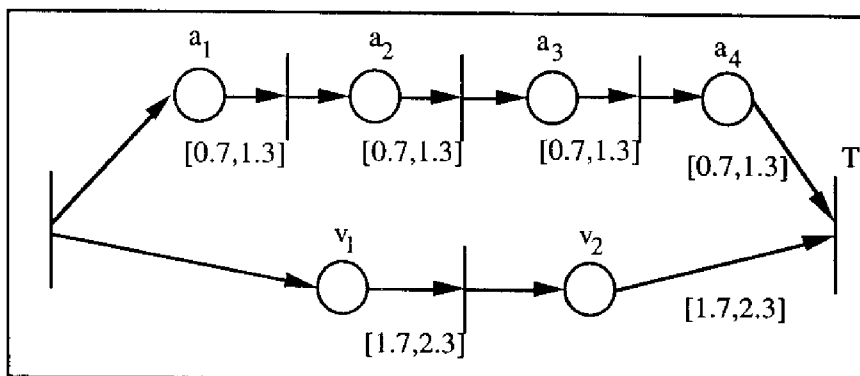


Figure I.3. Exemple d'ATPN

Sur cet exemple, appliquer la pire analyse possible montre que :

- si les trois premières places du flux audio se terminent au plus tôt (i.e. 0,7) alors l'intervalle dynamique de tir de la place a_4 est $[2,8; 3,4]$.

- si la première place du flux vidéo se termine au plus tard (i.e. 2,3), alors l'intervalle dynamique de tir de la place v_2 est [4; 4,6].

Dans ce cas, l'intersection des deux intervalles dynamiques est vide et les règles de tir [WAL83] ne résolvent pas ce problème (i.e. le modèle est incomplet et la présentation multimédia peut s'arrêter).

Ce modèle possède donc un bon pouvoir de modélisation, la spécification de la synchronisation inter-flux pouvant s'exprimer facilement par simple fusion de transitions. Toutefois, les règles de tir associées au modèle ATPN ne prennent pas en charge les problèmes de désynchronisation inter-flux ; il n'a pas de sémantique formelle.

Les autres modèles de synchronisation multimédia

D'autres travaux ont essayé de modéliser les contraintes de synchronisation de scénarios multimédias, et ce en utilisant des formalismes divers et variés. Toutefois, toutes les approches sont basées sur les trois concepts suivants [BLA96] [PER96] :

- la modélisation par les intervalles de présentation des objets ;
- la modélisation par les dates de présentation, appelée aussi modélisation par des axes temporels ;
- la modélisation par des points de référence, ou points de synchronisation.

Modélisation basée sur des intervalles

Cette forme de modélisation est basée sur les durées de présentation des objets du scénario de synchronisation. En fait, cette modélisation respecte les 13 schémas de synchronisation de Allen [ALL83] comme les OCPN présentés plus haut. De ce fait, ils souffrent des mêmes lacunes en terme de pouvoir d'expression car ils ne peuvent modéliser que des comportements idéaux des scénarios ; par exemple, il est impossible de modéliser le cas où deux flux terminent simultanément si on ne connaît pas la durée réelle de présentation de ces flux.

Respectant ce type de modélisation, [ATE96] a proposé un formalisme basé sur Timed CSP (Communicating Sequential Processing), une extension temporelle de CSP.

[HUA96] a proposé un modèle basé sur les machines à états finis étendues et le langage de spécification Estelle. La gestion du temps se fait par la clause $\text{delay}(d_1, d_2)$ qui ne rend une transition tirable que si elle a été sensibilisée pendant plus de d_1 et moins de d_2 unités de temps. Malgré l'indéterminisme apporté par cet intervalle temporel, le pouvoir d'expression n'est pas accru, et l'exemple de synchronisation de la fin de la présentation de deux flux par la méthode des intervalles reste impossible.

De même, [FIL93] [FIL94] attribue aux objets multimédias un profil temporel (formalisé par l'équivalent d'un type abstrait algébrique). Ce modèle est entièrement équivalent à un OCPN [PER96].

Modélisation basée sur des axes temporels

Le principe de cette modélisation consiste à rattacher toutes les unités d'information à un axe des temps qui est une abstraction du temps réel (comme pour la technique de synchronisation basée sur les estampilles temporelles). Toutefois, cette méthode ne permet pas de modéliser des présentations non déterministes.

C'est le cas par exemple dans Athena Muse Project [HOD89], QuickTime [DRU92] ou MHEG [ISO90] [ISO93] [BLA96] qui utilisent ce concept de ligne temporelle.

De même, [DAI94] attribue aux flux une signature temporelle qui représente la date de début de présentation de chacune des unités d'information du flux à présenter. Il est donc impossible de prendre en compte un quelconque asynchronisme du flux ou du système.

Dans cette famille de modèles, il est également possible de prendre des axes virtuels, et d'estampiller les unités d'information par rapport à ces axes virtuels. Un axe virtuel est un axe temporel sur lequel le système de coordonnées (origine, norme) peut

changer. Il est ainsi possible d'accélérer ou de ralentir une présentation, comme dans HyTime [ISO92] qui est construit sur ce principe.

Modélisation basée sur des points de référence

Avec cette méthode, les objets multimédias sont regardés comme des séquences d'unités d'information logiques avec des points de début, de fin et des sous-unités qui sont des points de référence. La synchronisation entre les objets se fait en reliant leurs points de référence [BLA92] [STE90]. Le problème avec cette méthode survient lorsqu'il faut détecter les inconsistances de présentation (par exemple le cas où deux médias atteignent leurs fins, mais à des moments différents, alors qu'ils auraient dû terminer ensemble). [SCH96a] utilise une solution basée sur les points de référence, avec le système de résolution des inconsistances de [STE90], mais ce mécanisme reste un mécanisme annexe qui n'est pas modélisé au niveau des liens entre les points de référence.

Solutions mixtes

Des solutions mixtes (i.e. basées sur deux des familles précédentes) existent. Par exemple, [PER96] propose un modèle basé sur les intervalles et sur les axes temporels. Toutefois, même si chacun de ces modèles tire profit de son association avec l'autre, ils ont tous les deux été conçus pour des présentations dont les durées et dates de présentation sont connues à l'avance. Ils ne peuvent donc pas prendre en compte l'indéterminisme d'un système asynchrone.

[RUS94] [COU94b] [COU96a] propose un modèle basé sur un axe temporel (avec des dates qui peuvent éventuellement appartenir à un intervalle sur cet axe temporel), et des points de référence exprimés par l'intermédiaire de dépendances causales. Il utilise le langage RT-Lotos (Real Time Lotos), une extension temporelle du langage de spécification Lotos, pour exprimer les contraintes de temps et de dépendances. Ce modèle permet donc de prendre en compte un certain niveau d'indéterminisme avec des dates glissantes dans un intervalle, et les points de synchronisation (dépendances croisées entre deux points de référence) permettent de régler le problème de la synchronisation entre deux flux lors de la terminaison non simultanée de ces deux flux. Toutefois, ce mécanisme ne peut pas régler le cas de fortes désynchronisations inter-flux et plus généralement le problème de la synchronisation inter-flux en environnement asynchrone ; en effet, pour synchroniser un flux audio avec des transparents (le flux audio étant plus important que le flux des transparents) [COU96a], il faut fixer une gigue δ maximale à la présentation de chaque transparent suffisamment grande pour éviter de supprimer de la présentation un transparent qui pourrait être en avance (ceci conduirait à une dégradation non nécessaire et plus importante que de laisser le transparent dépasser sa gigue maximale). Toutefois, un système asynchrone n'a pas de borne sur la gigue qu'il peut engendrer, et dans le cas général d'un système asynchrone, il n'est pas possible de fixer la gigue δ a priori.

1.3.3. Le modèle TSPN

A partir du panorama précédent, il apparaît que les modèles de synchronisation multimédia et en particulier les extensions temporelles des réseaux de Petri qui ont été proposées jusqu'alors n'offrent pas un bon pouvoir de modélisation et d'expression pour spécifier des scénarios de synchronisation dans une application multimédia.

Ces limitations observées au niveau des modèles existant ont conduit à proposer un nouveau modèle (basé sur les réseaux de Petri⁸), appelé TSPN (Time Streams Petri Net), offrant à la fois le pouvoir d'expression du modèle TPN et le pouvoir de

⁸Pour l'aspect graphique des réseaux de Petri qui permettent de mieux visualiser les caractéristiques de synchronisation d'un document multimédia.

modélisation du modèle ATPN. Le modèle TSPN [DIA93a][DIA93b] [SEN94] s'inspire du modèle ATPN auquel il ajoute le typage des transitions par des règles de tir de transitions inter-flux.

Ainsi, les TSPN utilisent des intervalles temporels sur les arcs sortant des places, ce qui permet à la fois de tenir compte du non déterminisme temporel des systèmes distribués asynchrones et de la variabilité des temps de présentation des objets multimédias. Les intervalles temporels dans un TSPN sont des triplets (x^s, n^s, y^s) appelés intervalles de validité temporelle, où x^s , n^s et y^s sont respectivement les temps du traitement de présentation minimal, nominal et maximal. Les durées nominales sont utiles pour calculer la dérive temporelle sur les arcs (par rapport à la durée nominale).

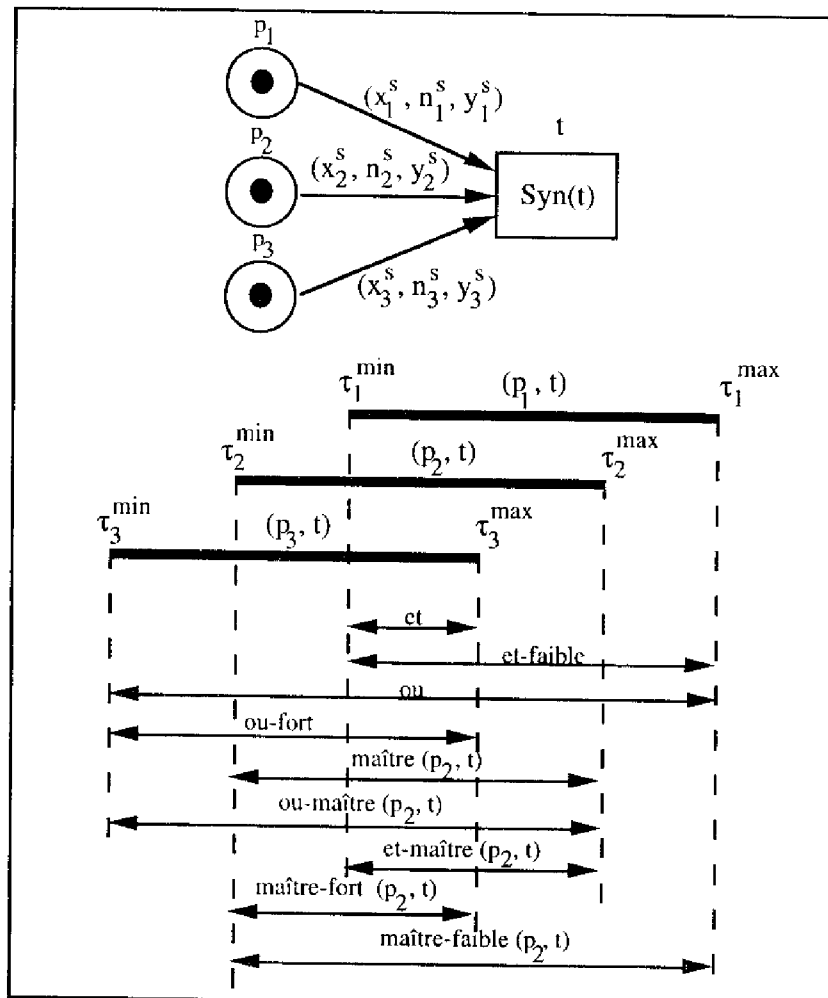


Figure I.4. Règles de tir des transitions inter-flux d'un TSPN

Les dérives temporelles entre flux multimédias peuvent être contrôlées de façon très précise grâce à 9 sémantiques de transition différentes. D'un point de vue exécution, ces sémantiques de synchronisation sont définies comme des instants de synchronisation, prenant en compte la durée réelle des processus; d'un point de vue modélisation, ces règles de tir définissent des intervalles de tir couvrant tous les instants de synchronisation possibles, obtenus par combinaison complète et consistante des intervalles dynamiques de validité temporelle des arcs concernés [DIA93a] [SEN94]. Par exemple, en utilisant ces règles de transition, il est possible de spécifier des mécanismes de synchronisation conduits par le processus le plus en avance (synchronisation de type "ou fort"), par le processus le plus en retard (synchronisation

de type "et faible") ou par un processus donné (synchronisation de type "maître"). Ces sémantiques de synchronisation permettent de définir les instants de synchronisation à partir d'un arc choisi statiquement ou dynamiquement. Ces neuf sémantiques de synchronisation et leurs intervalles de tir sont représentés sur la figure I.4, où, si τ_i est la date de sensibilisation de l'arc (p_i, t) , avec (x_i^s, n_i^s, y_i^s) comme intervalle statique de validité temporelle, alors τ_i^{\min} et τ_i^{\max} désignent $\tau_i + x_i^s$ et $\tau_i + y_i^s$ respectivement.

Un TSPN est défini comme suit :

Définition: Un TSPN est un tuple $(P, T, B, F, M_0, IM, SYN)$ où :

- (P, T, B, F, M_0) définit un réseau de Petri.
- IM est une fonction définie comme suit :
Soit A l'ensemble des arcs sortant des places : $A = \{a = (p, t) \in P \times T \mid B(p, t) \neq \emptyset\}$;
 $IM: A \rightarrow Q^+ \times Q^+ \times (Q^+ \cup \infty)$, $IM(a_i) \rightarrow (x_i^s, n_i^s, y_i^s)$
où le 3-uple associé à un arc représente les durées statiques minimale, nominale et maximale de présentation, respectivement; ces valeurs vérifient la relation :
 $0 \leq x_i^s \leq n_i^s \leq y_i^s$.
- SYN est une fonction de typage qui distingue les différentes règles de transition (i.e. les différentes sémantiques de synchronisations inter-flux) et définie comme suit :
 $SYN: T \rightarrow \{ \text{"ou"}, \text{"ou-fort"}, \text{"et"}, \text{"et-faible"}, \text{"maître"}, \text{"et-maître"}, \text{"ou-maître"}, \text{"maître-faible"}, \text{"maître-fort"} \}$

Chaque transition d'un TSPN est typée par une règle de tir ; cette caractéristique permet d'adapter facilement les règles de synchronisation aux attributs des médias à synchroniser et à la sémantique de l'application.

Les conditions de tir des transitions qui ont été représentées graphiquement sur la figure 5 peuvent s'exprimer de façon formelle. Ainsi, une transition peut être tirée à l'instant θ si et seulement si :

- la transition t est sensibilisée (i.e. tous les arcs entrants sont sensibilisés), et
- θ appartient, en fonction du type de transition, aux intervalles de tir suivants :

$$\text{si } SYN(t)=\text{et} \text{ alors } \theta \in [\max_{i \in I}(x_i), \max_{i \in I}(\min_{i \in I}(y_i)), \max_{i \in I}(x_i)]$$

$$\text{si } SYN(t)=\text{et-faible} \text{ alors } \theta \in [\max_{i \in I}(x_i), \max_{i \in I}(y_i)]$$

$$\text{si } SYN(t)=\text{ou} \text{ alors } \theta \in [\min_{i \in I}(x_i), \max_{i \in I}(y_i)]$$

$$\text{si } SYN(t)=\text{ou-fort} \text{ alors } \theta \in [\min_{i \in I}(x_i), \min_{i \in I}(y_i)]$$

$$\text{si } SYN(t)=\text{maître}(a_m) \text{ alors } \theta \in [x_m, y_m]$$

$$\text{si } SYN(t)=\text{ou-maître}(a_m) \text{ alors } \theta \in [\min_{i \in I}(x_i), y_m]$$

si $SYN(t)=et\text{-}maître(a_m)$ alors $\theta \in [\min(\max(x_i), y_m), y_m]$
 $i \in I$

si $SYN(t)=maître\text{-}fort(a_m)$ alors $\theta \in [x_m, \max(\min(y_i), x_m)]$
 $i \in I$

si $SYN(t)=maître\text{-}faible(a_m)$ alors $\theta \in [x_m, \max(y_i)]$
 $i \in I$

Où $I=\{i / a_i = (p_i, t)\}$ et a_m est l'arc maître.

L'accélération de flux :

La définition des règles de tir pour les transitions permet d'éviter les incohérences rencontrées dans les ATPN. En effet, il n'est pas nécessaire que l'intersection des intervalles dynamiques de validité temporelle soit non vide pour pouvoir tirer la transition, ce qui permet de définir toutes les formes de synchronisation inter-flux. Cependant, les règles de tir des transitions n'ont été définies que lorsque la transition est sensibilisée, i.e. lorsque toutes les places en entrée sont marquées. Or, dans un cas réel, il est tout à fait possible que l'un des flux soit très en retard, et que la place en entrée de la transition inter flux (P_2) ne soit pas marquée alors qu'un flux maître (P_4) (par exemple) atteint la borne supérieure de son intervalle de validité (figure I.5). Ce cas n'est pas défini par les règles de tir précédentes (c'est la même lacune que celle du modèle de [COU96a]). Il faut donc les étendre à ce cas là. Ceci a donc conduit à définir l'accélération de flux, et le modèle HTSPN (Hierarchical TSPN) [SEN96].

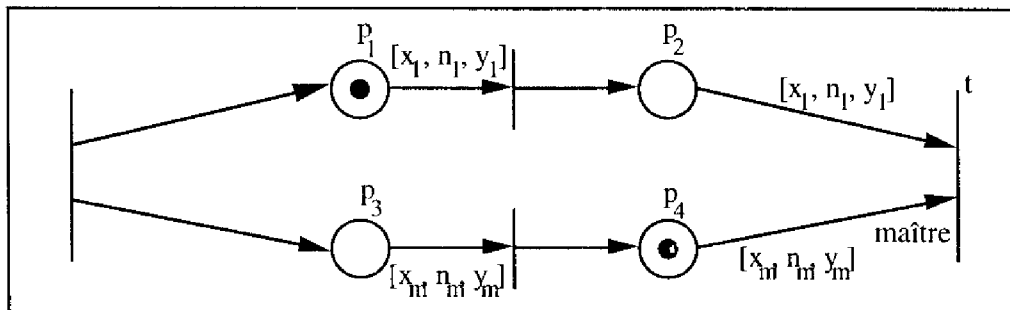


Figure I.5. Exemple d'inconsistance du modèle TSPN en cas de forte désynchronisation inter-flux

Au départ, le modèle HTSPN a été conçu pour modéliser les contraintes de synchronisation logiques et temporelles des documents hypermédias, mais il apparaît également idéal pour modéliser des scénarios de synchronisation multimédia dès lors que de fortes désynchronisations inter-flux peuvent apparaître.

Le principe des HTSPN consiste à utiliser une hiérarchie de TSPN pour modéliser des scénarios multimédias ou hypermédias. Ainsi, dans le cas de la modélisation de l'accélération de flux, deux niveaux de TSPN sont utilisés : une couche composite qui peut être vue comme modélisant la synchronisation inter-flux, et une couche atomique sous jacente qui modélise la synchronisation intra-flux. Ainsi, en traduisant le TSPN de la figure I.5, on obtient le HTSPN de la figure I.6, dans laquelle les places P et P' représentent les deux flux de présentation parallèles, chacun de ces flux étant ensuite décomposé en P1 et P2 pour P et P3 et P4 pour P' au niveau atomique. Il est, de cette manière, possible de modéliser l'accélération de flux : par exemple, dans le cas de la

figure I.6, lorsque la place P' atteint la borne supérieure de son intervalle de validité temporelle, alors que la place P_2 n'est même pas encore marquée, la transition t peut être tirée (car P est marquée). Cela conduit donc à la disparition du TSPN de niveau atomique associé à P (alors qu'il n'avait pas encore été complètement joué), et la présentation passe à la période suivante du scénario décrit par le TSPN de la couche composite : ceci correspond donc bien à une accélération du flux associé à P (qui était en retard). Pour de plus amples informations, une description formelle et complète du modèle HTSPN est donnée dans [SEN96].

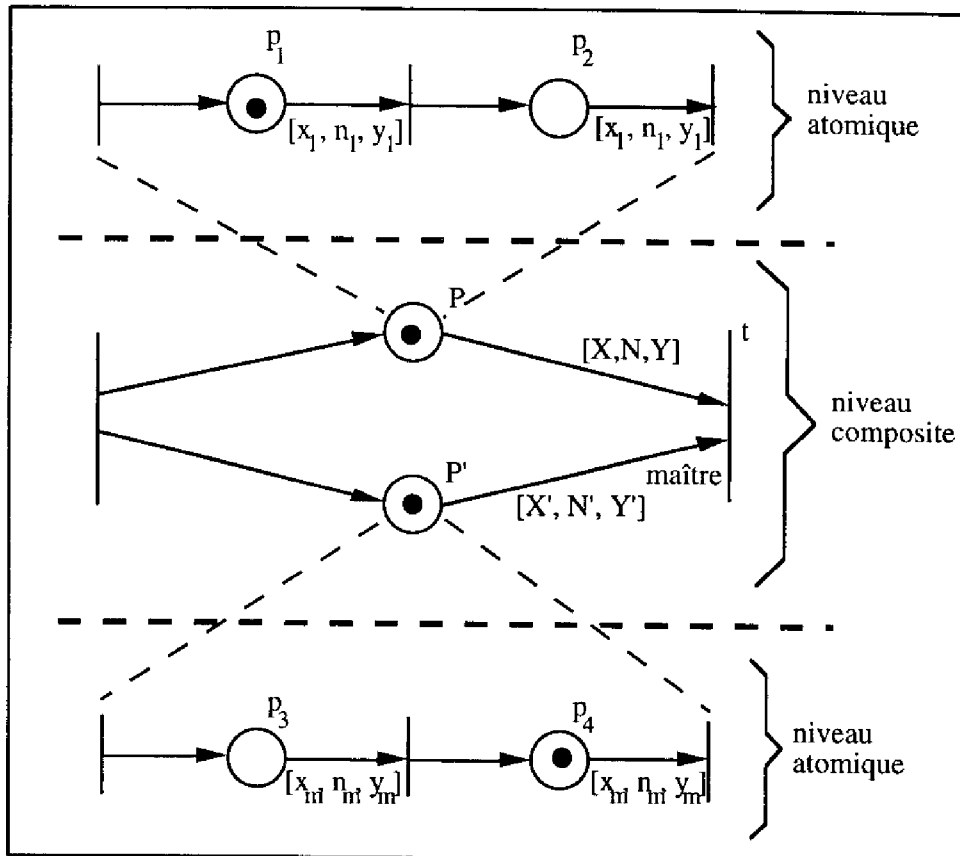


Figure I.6. Exemple de HTSPN

Remarques :

1. Les contraintes temporelles exprimées sur le TSPN de la couche composite ne sont pas forcément les mêmes que celles exprimées pour la couche atomique correspondante. Ainsi, $[X, N, Y]$ peut être différent de $[2x_1, 2n_1, 2y_1]$, si cela se justifie.
2. Le modèle HTSPN complet pour les documents hypermédias comporte 3 couches, avec une couche logique au dessus de la couche composite. Cependant, ici, pour des présentations multimédias, il n'y a pas besoin de ce troisième niveau, et seuls les deux niveaux bas sont utilisés.

Cependant, dans ce qui suit, le modèle TSPN sera utilisé pour modéliser la synchronisation des flux audio et vidéo d'une application multimédia. En effet, visuellement, la modélisation TSPN est plus agréable que la représentation HTSPN, car elle représente sur un même réseau de Petri toutes les contraintes de synchronisation et de parallélisme entre les objets des différents flux en les mettant face à face. Ainsi, même si le modèle TSPN n'est pas complet, il sera utilisé dans la suite (d'ailleurs, dans

les exemples qui suivront, le passage du TSPN au HTSPN sera trivial). De plus, sur un TSPN, l'accélération de flux peut être vue comme un simple saut de jeton (ce qui est une approche peu élégante du problème, mais très représentative). Aussi, l'accélération de flux pourrait être simplement prise en compte par les algorithmes qui seront proposés.

Enfin, dans les exemples qui vont suivre, et qui correspondent à nos études, avec les qualités de service la plupart du temps utilisées, ces phénomènes de fortes désynchronisations sont impossibles : il faudrait pour que de telles désynchronisations inter-flux apparaissent, tolérer sur les différents objets des giges très importantes, ce qui n'est pas courant, car les contraintes temporelles (en terme de gigue) sur les différents médias et sur l'audio et la vidéo en particulier sont en général faibles, pour que l'information présentée soit humainement compréhensible. Le modèle TSPN est donc suffisant pour représenter les contraintes de synchronisation de la visioconférence.

1.3.4. Utilisation des TSPN dans cette thèse

Le modèle TSPN constitue un modèle parfaitement adapté à la modélisation des contraintes de synchronisation de flux multimédias, en environnement asynchrone, pour les applications que nous considérons. Grâce à ses pouvoirs de modélisation et d'expression élevés, ce modèle permet de modéliser facilement des scénarios de synchronisation de complexité forte. De plus, à partir de la description en TSPN d'un scénario de synchronisation, il est possible de vérifier sa validité temporelle. Cette vérification se fait par l'intermédiaire de techniques conçues dans cette perspective, mais dont la description n'est pas l'objet de cette thèse⁹.

En ce qui concerne l'aspect synchronisation multimédia de cette thèse, le point essentiel consiste à étudier comment il est possible à partir d'un TSPN (ou HTSPN) de décrire le comportement d'une couche de synchronisation, et en particulier d'obtenir l'ordonnancement temporel des processus de présentation multimédia, de façon à ce que pour tous les flux de données, les contraintes de synchronisation intra et inter-flux soient garanties.

Cependant, ce qui caractérise un système asynchrone, c'est son imprévisibilité dans ses temps de traitement, et il n'est donc pas possible de déterminer l'ordonnancement avant l'exécution (i.e. de façon off line). Si le modèle OCPN est utilisé [LIT90] [LIT93], il est possible de déterminer un ordonnancement "off line". En effet, les durées associées aux places sont fixées, et donc l'ordonnancement des processus peut être déterminé avant l'exécution et sera le même lors de chaque exécution. Par exemple, avec l'OCPN de la figure I.7, le processus associé à la place p_1 débutera son exécution à la date 0, et le processus associé à la place p_2 débutera son exécution à la date t_1 , t_1 étant la durée du processus associé à p_1 .

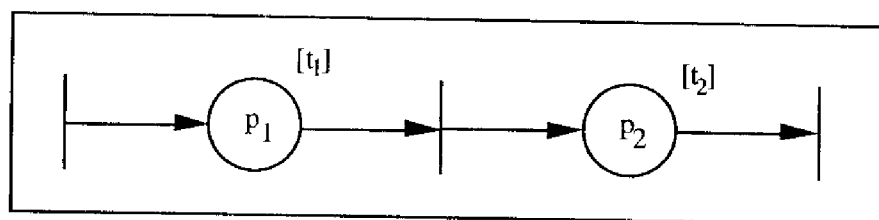


Figure I.7. Exemple d'OCPN

A cause de la variabilité temporelle des systèmes asynchrones, il n'est pas possible d'obtenir avec le modèle TSPN un ordonnancement "off line". En effet, la durée du traitement associé à une place est inconnue avant l'exécution, et ce n'est qu'au cours de l'exécution qu'il sera possible d'obtenir cette information. Par exemple, sur la figure I.8, le processus associé à la place p_1 débute à la date 0. Par contre le processus associé à p_2

⁹[COU96b] propose également une méthodologie pour vérifier la cohérence temporelle d'un document multimédia/hypermédia.

débutera entre $t_{1\min}$ et $t_{1\max}$, la date exacte n'étant connue qu'au moment de l'exécution, lorsque p_1 se termine.

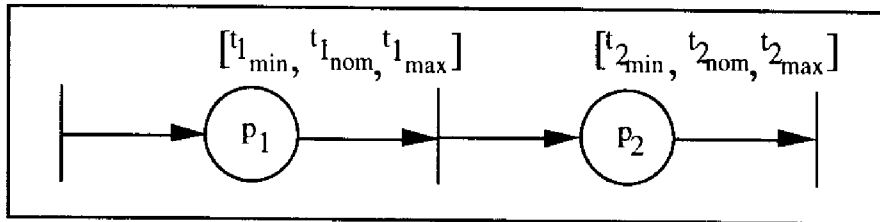


Figure I.8. Exemple de TSPN

Ainsi, avec le modèle TSPN, il n'est possible de n'inférer qu'un ordonnancement "on line"¹⁰.

Le modèle TSPN sert dans la suite de cette thèse à représenter et à implémenter les contraintes de synchronisation d'une application de visioconférence. La visioconférence illustre d'ailleurs très bien les possibilités du modèle TSPN.

Dans la visioconférence, un flux audio et un flux vidéo doivent être synchronisés. Un certain nombre de paramètres de qualité de service peuvent être représentés par un TSPN. Ainsi, la figure I.9 modélise une application de visioconférence dont les paramètres de qualité de service sont :

- le débit est de 10 images par seconde ;
- la gigue maximale acceptable sur un objet audio ou un objet vidéo est 10 ms [JEF94a] ;
- comme dans toute application de visioconférence, le son a un rôle prépondérant, la vidéo n'étant que secondaire ;
- la qualité de la synchronisation doit être excellente, c'est à dire que la dérive inter-flux ne doit pas excéder 100 ms [JEF94a], 100 ms étant la limite en deçà de laquelle le décalage audio vidéo n'est pas perceptible par l'homme.

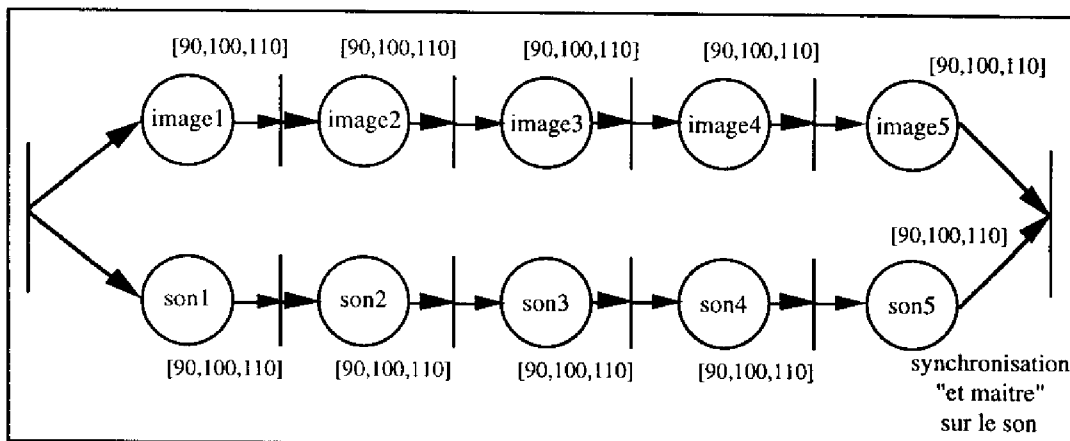


Figure I.9. Exemple de TSPN pour la visioconférence à 10 images par seconde

Les paramètres de qualité de service ont permis de déterminer les différents paramètres du TSPN représenté sur la figure I.9. Ainsi :

¹⁰Tout scheduling "off line" peut être déterminé de façon "on line", mais le contraire est impossible. La classe des scheduling que l'on peut déterminer de manière "on line" est donc plus vaste que la classe des scheduling que l'on peut déterminer de manière "off line".

- Le débit de 10 images par seconde a permis de déterminer la durée de présentation nominale d'un objet vidéo, i.e. 100 ms (si on considère des granularités identiques sur les flux audio et vidéo, c'est aussi le temps de présentation nominal d'un paquet son) ;
- La gigue maximale acceptable a permis de déterminer les intervalles de validité temporelle qui sont donc [90, 100, 110] ;
- La synchronisation inter-flux est de type "et-maître" sur le son. En effet, le son ayant une prépondérance par rapport à la vidéo, ses contraintes de synchronisation intra-flux doivent être absolument respectées, même si celles sur la vidéo ne le sont pas. Cependant, il faut synchroniser l'un par rapport à l'autre deux flux continus, et éviter au maximum les discontinuités sur le flux vidéo (pouvant être causées par le mécanisme de l'accélération de flux). Pour cela, la règle de tir "et-maître" a été choisie, assurant ainsi que les contraintes sur le son seront garanties, mais essayant autant que possible de respecter aussi les contraintes temporelles sur la vidéo, lorsque c'est possible ;
- La dérive inter-flux ne doit pas dépasser 100 ms, aussi la période de synchronisation inter-flux correspond à la présentation de 5 images. En effet, la dérive maximale sur 5 objets audio ou vidéo est de 50 ms. La dérive entre les deux flux est donc au maximum de 100 ms.

Si par contre, la qualité de service de la visioconférence n'est que d'une image par seconde (les autres paramètres restant les mêmes que pour l'exemple de la figure I.9), le TSPN représentant les contraintes de synchronisation est donné sur la figure I.10 ; l'intervalle de validité temporelle d'une image vidéo est [990, 1000, 1010], l'intervalle de validité temporelle d'un son devient [190, 200, 210] si la période de synchronisation inter-flux correspond toujours à la présentation de 5 sons ; cette période engendre une borne maximale de la dérive inter-flux de 60 ms ce qui est inférieur aux 100 ms demandées. Toutefois, la période de synchronisation ne peut pas correspondre à la présentation de 2 images vidéo (10 sons) car la dérive inter-flux pourrait atteindre 120 ms, ce qui n'est pas acceptable. Par contre, étant données les faibles contraintes temporelles sur le flux vidéo (des durées de l'ordre de la seconde ne sont pas réellement contraignantes en informatique), la synchronisation inter-flux peut être assouplie par rapport au flux vidéo. Ainsi, une synchronisation inter-flux de type "maître" sur le son serait satisfaisante, car seules les contraintes sur le son peuvent être difficiles à garantir.

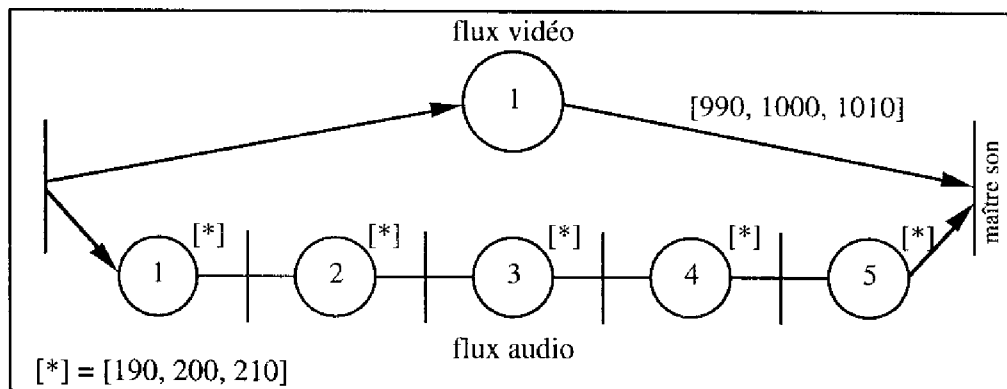


Figure I.10. Exemple de TSPN pour la visioconférence à 1 image par seconde

La figure I.11, quant à elle, modélise les contraintes de synchronisation d'une application de visioconférence dont la qualité de service est très forte. Ses caractéristiques sont les suivantes :

- le débit vidéo est de 20 images par seconde, et la gigue maximale acceptable est de 10 ms sur chaque objet vidéo. L'intervalle de validité temporelle sur une image est donc [40, 50, 60] ;
- le son est stéréophonique et se compose donc de deux flux (le flux droit et le flux gauche). La gigue maximale tolérable sur un objet sonore est 1 ms et la dérive inter-flux entre les deux flux audio ne doit pas dépasser 10 ms : ces contraintes engendrent ainsi un intervalle de validité temporelle des sons de [9, 10, 11], et la synchronisation inter-flux entre les deux flux audio correspond à la présentation de 5 objets sonores ;
- la dérive inter-flux audio / vidéo de 100 ms contraint à réaliser une synchronisation inter-flux toutes les 6 images (30 sons) : la dérive inter-flux maximale possible est donc de 90 ms (< 100 ms) ;
- enfin, comme pour le cas de la figure I.9, la synchronisation inter-flux audio / vidéo pourrait être de type "et-maître" sur les flux du son stéréo (pour les raisons déjà évoquées). Par contre, étant données l'importance des deux flux audio et la précision de leur contraintes, on peut choisir de ne pas défavoriser l'un des deux flux, et donc de réaliser des synchronisations de type "et" entre les deux flux : cette synchronisation favorise le flux le plus en retard, et aucune information sonore ne sera écartée.

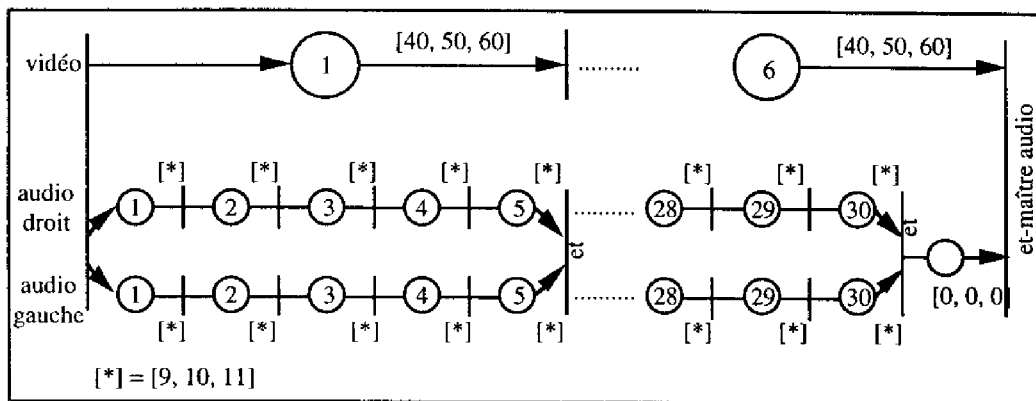


Figure I.11. Exemple de TSPN pour la visioconférence stéréophonique à 20 images par seconde

Les TSPN des figures I.9, I.10 et I.11 permettent donc de bien représenter les contraintes de présentation de l'application de visioconférence. Cependant, de nombreux autres paramètres de qualité de service doivent être pris en compte (en plus des paramètres de présentation) par l'application. Le chapitre 2 présente d'ailleurs en détail tous les mécanismes de synchronisation et de respect des contraintes de qualité de service de l'application de visioconférence qui correspond à cette approche : PNSVS (Petri Net Synchronized Visioconference System).

2. La coopération

2.1. Introduction sur la coopération et le travail coopératif

La notion de travail coopératif est ancienne. Cependant, son application au travail sur ordinateur est assez récente et a commencé à s'étendre vers le milieu des années 80.

2.1.1. CSCW et collecticiel

Le terme **CSCW** (*Computer Supported Cooperative Work*) ou **TCAO** (*Travail Coopératif Assisté par Ordinateur*) a été introduit pour désigner tous les travaux et

toutes les applications développées pour des groupes d'utilisateurs [BAN89] [BLA94]. Le but de ces applications n'est pas seulement de prendre en compte les interactions entre une personne et un ordinateur, mais également les interactions entre un groupe de personnes au travers d'un réseau d'ordinateurs. Ici, il n'y a pas uniquement un dialogue homme/machine mais également des échanges entre des entités (individus ou agents) via un système informatique. Le CSCW est donc un domaine pluridisciplinaire dans lequel interviennent des recherches provenant des sciences sociales (sociologie, théorie des organisations, psychologie) qui prennent en compte l'organisation des personnes travaillant ensemble, leurs rapports, l'efficacité d'un groupe pour la réalisation d'un travail, etc. ... Les sciences cognitives interviennent aussi dans le CSCW. Leur apport se situe dans l'interprétation de la sémantique des informations échangées, l'attribution des tâches, la planification, l'aide à la réalisation de tâches en commun... En dernier lieu, le CSCW utilise les travaux réalisés en informatique dans les domaines des systèmes distribués et des télécommunications pour les stockages, transferts et échanges d'informations. C'est ce dernier point qui nous intéresse en tout premier lieu dans cette thèse.

Le CSCW peut être découpé en quatre grandes catégories de complexités conceptuelles différentes [VIL95] :

Le Travail Collectif : cette activité concerne un ensemble de personnes qui partagent équitablement les responsabilités et les résultats lors de la réalisation d'une tâche. Tous les membres du groupe ont les mêmes pouvoirs, les mêmes responsabilités et ils voient les mêmes résultats que les autres membres du groupe ; certaines conférences sont des cas classiques de travail collectif.

La Coopération : un groupe coopératif comporte trois niveaux. Le premier niveau décrit les buts, les tâches et les objectifs à atteindre, exprimés de telle sorte qu'ils soient compréhensibles et interprétables par les membres du groupe. Le second niveau est un méta-niveau de décomposition et d'attribution des tâches aux membres d'un groupe ; à ce niveau, l'accent est mis sur la façon de réaliser les tâches, la décomposition des tâches en sous-tâches et la répartition des sous-tâches entre les différents membres du groupe, et ce en évitant le plus possible les conflits, désaccords et redondances... Le troisième niveau comprend l'exécution des tâches par le groupe avec la nécessité que les membres se synchronisent et aient des résultats qui convergent. Bien entendu, pendant la réalisation des tâches, des désaccords ou des conflits peuvent survenir. Il faut donc des mécanismes pour résoudre ces problèmes afin que les membres du groupe n'aboutissent pas à des vues divergentes, voire contradictoires [DUR87]. Un groupe coopératif peut donc être vu comme un groupe qui partage les responsabilités pour réaliser des tâches en commun.

La Collaboration : elle concerne un ensemble d'individus qui possèdent chacun une certaine vue d'un problème et cherchent à converger vers un même but global, chacun participant à la tâche globale et effectuant une partie de la résolution du but. Cette catégorie est caractéristique des solveurs distribués de problèmes dans lesquels un ensemble d'agents cherche à résoudre de façon coopérative un problème global [DUR89]. Ce groupe d'agents a en commun une tâche à réaliser et ils partagent la réalisation de cette tâche.

Les Mondes "multi-agents" : cette extension des catégories précédentes considère un ensemble de groupes d'utilisateurs qui évoluent dans un même monde [HEN89] [HEW91]. Cela complique les problèmes rencontrés car on ne considère plus seulement un groupe d'utilisateurs mais un ensemble de groupes en interaction. Un utilisateur (ou agent) peut faire partie de plusieurs groupes. Comme décrit dans les catégories précédentes, des problèmes de cohérence et de gestion de conflits à l'intérieur des groupes sont rencontrés, mais ces problèmes se posent également dans les échanges inter-groupes.

Le terme CSCW englobe l'ensemble des études concernant le travail en groupe assisté par ordinateur, ceci comprenant les études humaines et sociologiques. Le collecticiel (ou groupware) désigne par contre les logiciels conçus pour supporter des groupes d'utilisateurs. Il désigne ainsi tous produits logiciels qui supportent les applications du CSCW [MIT91] [ELL91] [KAR94a].

2.1.2. Coopération synchrone, coopération asynchrone

En plus de ces quatre catégories de CSCW, il existe d'autres caractéristiques qui les distinguent ; [VIL95] donne plusieurs critères pour les séparer, et en particulier, il distingue la coopération synchrone de la coopération asynchrone.

La coopération synchrone comprend les applications qui requièrent la présence simultanée des membres du groupe pour effectuer le travail coopératif. Elle s'appuie sur des supports de communication dont le délai de bout en bout est suffisamment court pour permettre des interactions de type face à face. Quelques exemples de ce style d'application sont les conférences [OHM92], la télé-présence [GUI92].

La coopération asynchrone, au contraire, correspond aux applications coopératives qui travaillent en différé, de type courrier électronique. Pour se dérouler, la coopération asynchrone ne nécessite pas la co-présence des différents membres du groupe.

Parfois, il est difficile de classifier les applications coopératives en synchrones ou asynchrones [ROD92]. Par exemple, l'édition de textes en commun peut se faire de manière synchrone ou asynchrone [MIN93]. Il est également possible que des applications coopératives utilisent les deux modes.

2.2. Caractéristiques des applications coopératives

Cette section présente un découpage modulaire des besoins des systèmes coopératifs (figure I.12). Ce découpage peut servir de base à la réalisation d'un modèle plus conséquent pour décrire des systèmes coopératifs. Il s'inspire essentiellement des modèles de [SMI89] [ROB94] et des idées de [BAN89] [KRA91] [WIL91]. On peut également retrouver ce découpage dans [VIL95].

2.2.1. Schéma de découpage

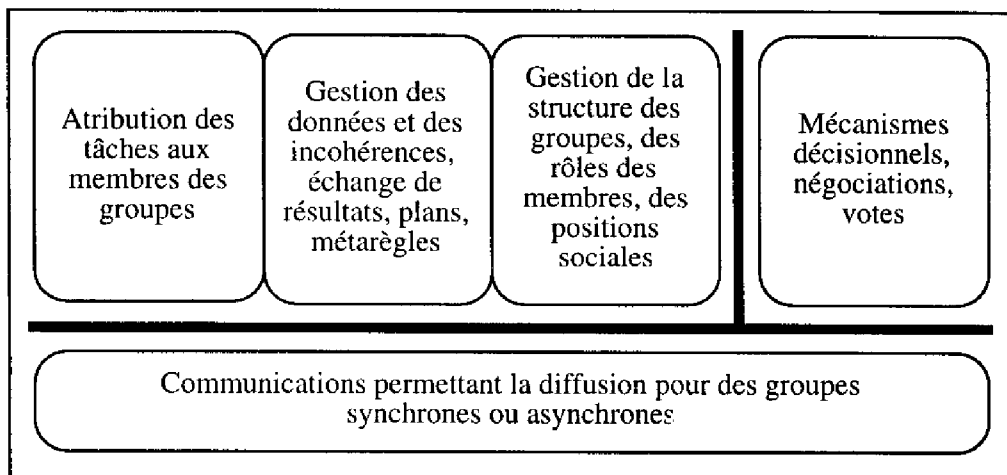


Figure I.12. Les principales fonctionnalités d'un système coopératif

Les groupes fonctionnels de gestion des tâches, des données, de la structuration des groupes et des mécanismes décisionnels doivent s'appuyer sur des services de communication permettant aux entités en coopération d'échanger des informations. Les modules de gestion des tâches, des données, et des groupes sont finalement fortement reliés entre eux, et sont même inter-dépendants. L'exécution d'une tâche donnée introduit des manipulations d'informations et dépend de la configuration et des positions des agents dans le groupe coopératif. Les mécanismes décisionnels peuvent intervenir et être utilisés par les trois modules précédents soit pour des prises de décision, soit pour gérer des conflits.

La suite présente chacun de ces groupes fonctionnels, mais s'intéresse plus particulièrement aux communications, à la gestion de la structure des groupes et aux mécanismes décisionnels. La gestion des tâches et des données n'a été que très peu abordée dans cette thèse.

2.2.2. Communications

Les notions de communication de groupe et de gestion de groupes de communication proviennent directement des domaines des systèmes distribués et des réseaux.

Pour des communications entre les membres de groupes asynchrones, les messageries de type X.400 ont été utilisées avec la possibilité de se servir du répertoire X.500 pour nommer et stocker les membres des groupes ou les ressources rattachées aux groupes [PRI89] [BEN93].

Comme exemples de réalisations pour des groupes synchrones, les systèmes d'exploitation V [CHE85] [CHE88], AMOEBA [KAA91] et ISIS [BIR87] [BIR91] [SCH93] fournissent des facilités pour les communications de groupes, et pour la gestion d'adresses de groupes. Dans ces trois systèmes, les agents peuvent créer dynamiquement des groupes qu'ils peuvent joindre et quitter, et peuvent échanger des informations dans ces groupes. Les communications de groupe proposées pour lesquelles les agents peuvent émettre, recevoir des données ou les deux, ont ainsi conduit à la production d'un document ISO pour les transmissions de données multipoints [ISO2031]. Les services proposés pour les communications de groupes synchrones [HUG89] [LIA90] [NGO91] [OWE93] diffèrent suivant la fiabilité acceptée, l'ordonnancement entre les messages, l'ordonnancement entre plusieurs diffusions provenant de sources différentes, le nombre d'acquiescements attendus...

2.2.3. Structuration des groupes

La définition de la structure des groupes, des rôles des membres et des positions sociales est tirée des travaux de la sociologie, des études du travail en entreprise et de l'automatisation des tâches de bureautique. Par exemple, un groupe de personnes peut être organisé en une hiérarchie ou bien en une organisation démocratique.

[HEW91] propose un modèle de groupe basé sur le modèle d'acteurs dans lequel les acteurs sont regroupés en organisations. On peut ainsi avoir des communications entre acteurs appartenant à la même organisation ou entre organisations différentes.

[HEN89] et [BEN93] effectuent un découpage en domaines pour la gestion et le partage des informations. Ces domaines permettent la gestion des ressources et des informations rattachées à chaque domaine, des rôles et des personnes de chaque domaine.

La structuration du groupe coopératif permet aux agents de posséder des connaissances sur eux-mêmes et sur les autres agents avec lesquels il coopèrent [KRA91] [DUR89]. Cette structuration introduit des bornes ou des limites d'action de chaque agent à l'intérieur des groupes coopératifs [DUR87]. Ainsi, suivant la structure d'un groupe, un membre peut ne pas connaître l'ensemble des membres du groupe et peut n'être en relation qu'avec un sous-ensemble des membres.

Le ou les rôles de chaque utilisateur servent à spécifier les responsabilités et les devoirs de chacun. Un rôle offre ainsi un ensemble de possibilités, de connaissances, de qualifications [SMI89] [DOL89]. Les rôles définissent des capacités de raisonnement qui s'appliquent sur ces données et connaissances. Ces capacités permettent de filtrer, d'interpréter, d'analyser les données et connaissances échangées [DUR87]. La structuration du groupe, en mettant en place les liaisons entre les divers membres du groupe, permet aussi de choisir les types de coordination entre agents et d'attribuer les rôles et fonctions des agents dans la structure logicielle. Suivant leur position dans le groupe, certains agents peuvent avoir un rôle coordinateur ; des autorités et pouvoirs peuvent leur être attribués afin de surveiller l'avancée du travail, de contrôler et d'arrêter les déviations des agents, de supprimer d'éventuels blocages ou ralentissements inacceptables. Ces capacités de raisonnement permettent de planifier le travail, d'éviter ou de résoudre d'éventuels conflits.

Les groupes servent aussi à maintenir une connaissance commune à un ensemble de personnes [BIG89], [HEW91]. Un groupe coopératif, lorsqu'il évolue, crée un contexte qui lui est propre et, suivant l'apport ou la mise à jour de nouvelles informations, ce contexte est modifié dans le temps. A chaque instant, ce contexte définit la connaissance actuelle du groupe, connaissance dépendante de l'historique du groupe et des informations échangées. Ce contexte doit être maintenu, notamment lorsque le groupe coopératif est dynamique et qu'il peut recevoir de nouveaux arrivants. Il faut donc définir des points pour lesquels le contexte courant est cohérent, afin de pouvoir le stocker ou le communiquer.

Les groupes coopératifs peuvent donc être définis par une structure statique, figée tout le long de la réalisation du travail coopératif, ou bien par une structure plus dynamique qui cherche à s'adapter à chaque instant aux besoins instantanés du travail à réaliser. Cette structure dynamique peut être guidée par les buts ou sous-butts associés au groupe, qui représentent les travaux que ce groupe doit effectuer.

2.2.4. Gestion des données

La gestion de la signification des données et des incohérences est un problème souvent rencontré. Ces incohérences proviennent du fait qu'un utilisateur ne possède qu'une vision partielle de l'ensemble du système, n'accède qu'à une partie des données et doit donc se contenter pour raisonner des données qu'il peut connaître.

Par exemple, lorsque un système se trouve découpé en plusieurs groupes, un utilisateur ne peut pas avoir accès aux informations appartenant aux groupes dont il ne fait pas partie. Par conséquent, le découpage en plusieurs groupes masque de l'information aux agents ne faisant pas partie des groupes et est donc un facteur supplémentaire d'incohérence [BIG89]. Un autre facteur d'incohérence est l'arrivée de nouveaux membres dans un groupe. En effet, les nouveaux arrivants doivent être mis au courant de l'évolution du groupe pour qu'ils aient la même information que les autres membres [BIG89]. Le système doit donc garder l'état courant du groupe afin de l'indiquer aux nouveaux arrivants.

Une première technique utilisée pour résoudre le problème des incohérences ou conflits est la structuration des groupes [DUR87]. Par exemple, si les responsabilités sont partagées entre les membres du groupe, chaque membre s'occupe de la partie qui lui a été confiée. On peut avoir un agent coordinateur qui supervise les autres agents. Dans une hiérarchie, les agents des niveaux supérieurs ont une vue plus complète du problème à résoudre et sont donc plus aptes à gérer les conflits que les agents des niveaux inférieurs qui n'ont que des vues très partielles des problèmes. Une deuxième technique est l'ajout de "plans" à chaque agent pour le guider dans sa démarche [DUR87]. Cette idée se rapproche assez de celle des rôles attribués à chaque personne. On peut aussi ajouter des métarègles [DUR87] pour tester l'avancée des autres membres du groupe et pour se synchroniser avec eux. Il est également possible pour lever les conflits que les agents échangent entre eux des résultats partiels [DUR87] [DUR89], ou bien qu'ils échangent entre eux les données manquantes [INT91].

2.2.5. Attribution des tâches

L'attribution des tâches aux membres des groupes consiste, lorsque le travail à réaliser a été découpé en plusieurs tâches, à répartir chacune d'elles entre les différents membres du groupe. Cette répartition doit être la meilleure possible de façon à ce que les tâches soient attribuées aux membres les plus à même de les effectuer [SMI81]. Il faut également réduire au maximum les conflits pouvant apparaître entre les tâches [ZLO91]. D'autres règles peuvent régir l'attribution des tâches, mais elles ne seront pas abordées car elles sont spécifiques au type de coopération dans l'application considérée.

2.2.6. Mécanismes décisionnels

Les mécanismes décisionnels interviennent lorsque les utilisateurs d'un système coopératif doivent prendre une décision entre eux.

Ces mécanismes peuvent être informels sous forme de discussions [HAH89], ou bien peuvent faire appel à des mécanismes plus formels tels que les votes [HAH89] [CHE90] [AHA91] [KUM91] [NEI92]. Une décision peut être laissée à la charge d'un arbitre [HEW91].

En fait, pour résoudre des conflits, les agents regroupés en organisations forment des forums [HEW91]. Un forum correspond à une assemblée où les agents négocient et délibèrent entre eux. Un forum possède une politique propre pour prendre une décision, politique qui peut être un vote ou bien un arbitrage par un agent...

2.3. Modélisation des activités coopératives

Ainsi, il apparaît dans ce qui précède que plusieurs possibilités sont possibles quant à la structuration des groupes, les décisions qui peuvent être prises, le moyen de les prendre, etc. et il est donc intéressant de modéliser ces paramètres de coopération. La suite présente donc d'abord les principales approches de modélisations rencontrées dans la littérature, puis le modèle que nous avons utilisé pour modéliser la structure dynamique des groupes coopératifs et les échanges de données (communications), qui sont les deux points sur lesquels l'essentiel de nos efforts s'est porté.

2.3.1. Les modèles existants

2.3.1.1. La logique

La logique modale peut être utilisée pour modéliser des systèmes distribués [MAZ91]. Les formules logiques permettent de représenter les propriétés globales du système et les propriétés locales de chaque agent. L'ensemble des propriétés que peut posséder un système se trouve ainsi défini par des formules logiques. La logique modale - appliquée à la modélisation de protocoles de validation négociée (comme le protocole "contract net") et à des systèmes basés sur les tableaux noirs - a permis de montrer que ces protocoles sont parfois irréalisables sous certaines hypothèses telles que les fautes ou les pertes de messages [MAZ91].

L'utilisation de la logique du premier ordre dans le système DARES [INT91] permet de représenter les messages échangés. Ce système est un résolveur distribué de problèmes [INT91]. Un problème est en fait un but à prouver, représenté au moyen de formules logiques du premier ordre. Le problème est résolu lorsque les formules logiques qui le représentent forment un théorème. Les clauses à prouver sont réparties entre les agents qui travaillent de façon coopérative. Lorsqu'un agent se trouve bloqué et ne peut plus résoudre de clause, des heuristiques peuvent s'appliquer. Elles visent à fournir de la connaissance supplémentaire à l'agent qui se trouve bloqué. La logique du premier ordre peut aussi servir à la vérification de propriétés [KRE89].

2.3.1.2. Réseaux de Petri

Les réseaux de Petri sont utilisés pour la planification de tâches bureautiques coopératives [VIC89]. Le réseau est simulé et vérifié. Les conflits sont détectés et signalés par le système. Les réseaux de Petri étendus servent très souvent à modéliser et analyser la circulation de documents pour des tâches bureautiques au sein de groupes coopératifs [TOU94] [AGO94] [AAL94] [KEL94].

Les réseaux de Petri peuvent aussi être employés pour représenter les plans individuels de chacun des agents d'un groupe d'agents. Ces agents évoluent dans un environnement commun, mais ne sont pas forcément au courant de tout ce que font les autres agents. Pour éviter des incompatibilités, des redondances entre les différents plans établis, il est nécessaire de les associer et de les coordonner. Le système fusionne les réseaux représentant chaque agent, analyse les différents plans, lève les contraintes, les incompatibilités et résout les problèmes afin d'obtenir un plan global synchronisé.

2.3.1.3. Acteurs

Le modèle acteur est aussi utilisé pour représenter des agents coopératifs [HEW91]. Cependant, ce formalisme n'est pas suffisant pour représenter des groupes coopératifs. Les acteurs sont donc rassemblés dans des groupes appelés "Organisations of Restricted Generality" (ORG). Chaque acteur est placé dans un seul ORG. Les communications entre deux groupes se font via des acteurs spéciaux qui servent d'interface entre les groupes. Ainsi, un acteur d'un groupe peut communiquer avec des acteurs d'autres groupes. Lorsque des conflits surviennent, les acteurs forment des forums qui possèdent plusieurs politiques de négociation (votes, arbitrage par un chef...).

Un modèle proche du modèle d'acteurs permet également une autre représentation des agents de groupes coopératifs [MAR90]. Un agent est composé d'une file d'attente qui permet de stocker les messages reçus et d'un interpréteur de messages qui sélectionne les messages de la file suivant certains critères sémantiques et non suivant leur ordre d'apparition dans la file. Cet interpréteur retire les messages importants de la file et invoque les méthodes adéquates. De plus, les agents possèdent un comportement propre représenté sous la forme d'une machine à états finis. De ce fait, les agents sont appelés semi-autonomes car leurs actions ne sont pas seulement conditionnées par les messages reçus comme pour des acteurs classiques, mais aussi par leurs états internes. Afin de manipuler des ensembles d'agents, un modèle de groupe a été défini. Une organisation de grande taille est définie comme des ensembles de groupes d'agents. Un agent peut ainsi envoyer un même message vers un groupe d'agents.

2.3.2. Un modèle de coopération basé sur le partage d'informations

Pour finir l'évocation des modèles existants, il existe des systèmes qui s'appuient sur un modèle de partage d'information [BEN92] ou de partage de documents [FUK94] plutôt que sur un modèle procédural d'échange d'informations par envoi de messages. Dans le projet Grace [BEN92], la communication est représentée par l'accès à des informations structurées à l'intérieur de domaines qui définissent des espaces spécifiques d'activités. On n'a plus d'envoi de messages vers un groupe de gens car on se contente de rendre un document accessible et visible au groupe.

C'est cette approche qui a été retenue dans le cadre de cette thèse, et la suite décrit le modèle qui en découle.

2.3.2.1. Modèle de systèmes coopératifs

Dans ce contexte, le modèle de coopération le plus faible pour un agent signifie la mise à disposition d'autres agents, d'une partie de ses connaissances, de ses informations personnelles et de ses données propres. En conséquence, par définition, un agent coopère avec un autre agent s'il rend une partie de ses données visibles et disponibles pour l'autre agent ; il y a distribution explicite [DIA92], [DIA93e], [DIA94c].

Soit A un ensemble d'agents. $A = \{A_i\}$. Chaque agent A_i possède un ensemble d'informations et de connaissances dont il est propriétaire. Soit $P = \{p_i\}$ l'ensemble des prédicats représentant la connaissance et les informations propres à un agent donné A_i . Une partie de ces informations reste uniquement connue de cet agent. Elle forme les informations locales à cet agent. Par contre, une autre partie des données va comprendre les données exportées qui vont être mises à disposition d'autres agents. Dans chaque agent, les prédicats sont soit locaux, soit exportés. Lorsqu'un agent va faire connaître à d'autres agents ses données exportées, alors il sera en relation de coopération avec ces autres agents. Le partage d'information crée une relation R entre les divers agents coopérants (figure I.13). Cette relation de partage d'information introduit donc une structuration de l'ensemble des agents du groupe coopératif. Les liens définis par les prédicats exportés par un agent vers d'autres agents caractérisent la relation de coopération. Ceci nous conduit aux définitions suivantes :

- **Définition** : A_i est en coopération avec A_j lorsque A_i permet à A_j d'accéder aux valeurs de certaines de ses informations exportées. Ces informations deviennent connues de A_j .
Les prédicats (ou informations) exportés par un agent deviennent connus des agents qui précèdent cet agent suivant la relation R , et non connus directement par leurs antécédents. Les prédicats locaux ne sont pas connus des agents qui sont en coopération avec l'agent considéré.

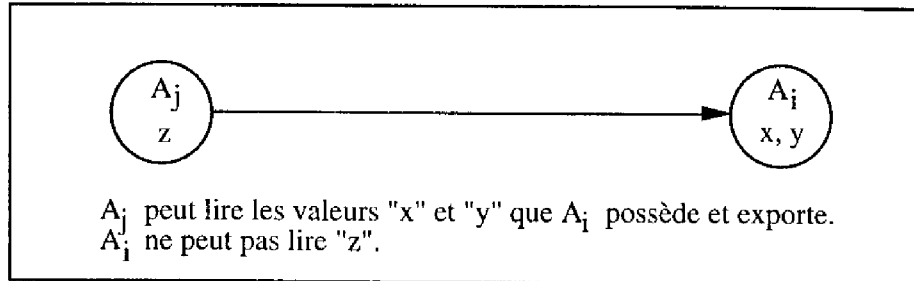


Figure I.13. Le modèle de coopération basé sur le partage de données

La relation R de partage d'information peut être représentée par un graphe. Les sommets représentent les agents en coopération, les flèches décrivent la relation R . Les sommets contiennent les informations et les connaissances de chacun des agents, les flèches du graphe représentent l'accès aux valeurs des données exportées par les agents. Le graphe formé de l'ensemble des agents coopérants et de la relation R est appelé "graphe de coopération" (figure I.14).

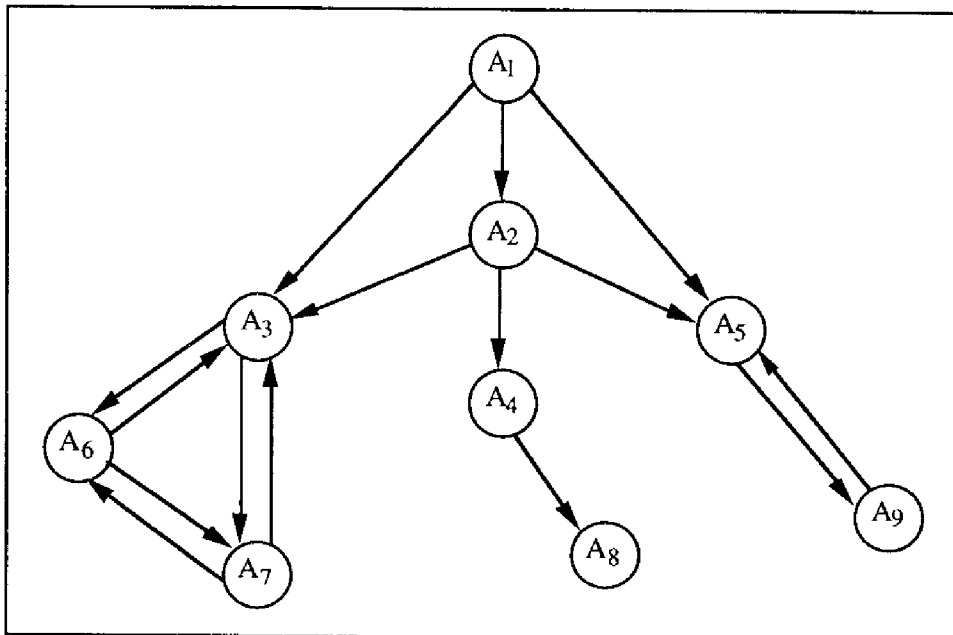


Figure I.14. Un graphe de coopération

La relation R peut posséder diverses propriétés :

R est réflexive, donc $A_i R A_i$, car un agent connaît ses propres prédicats.

Dans le cas général, R n'est ni symétrique ni transitive (figure I.14). Si R est symétrique, la coopération est dite bilatérale (figure I.15 (a)). Si R est transitive, la coopération est hiérarchique (figure I.15 (b)). Si R est à la fois symétrique et transitive, on dit qu'elle est parfaite ou totale (figure I.15 (c)). Les sémantiques de la coopération

qui définissent le comportement d'un agent donné vont ainsi dépendre de la relation R , i. e. de la structure de la coopération.

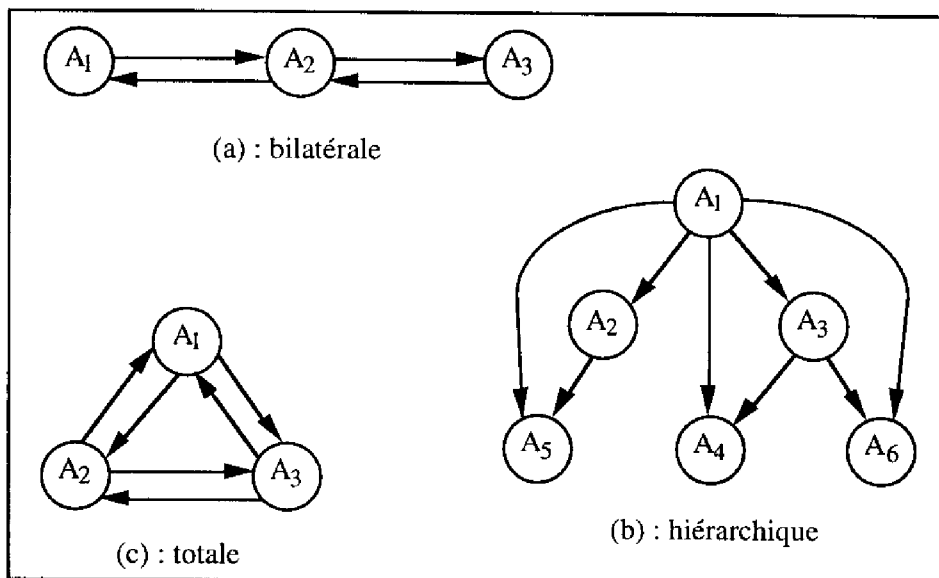


Figure I.15. Quelques structures de coopération particulières

2.3.2.2. Dépendances des données et prédicats

La relation R introduit une structuration dans le groupe coopératif. Cette structuration entraîne également des dépendances entre les prédicats de chacun des agents A_i , un agent pouvant être propriétaire de prédicats dont la valeur dépend d'autres prédicats exportés par d'autres agents.

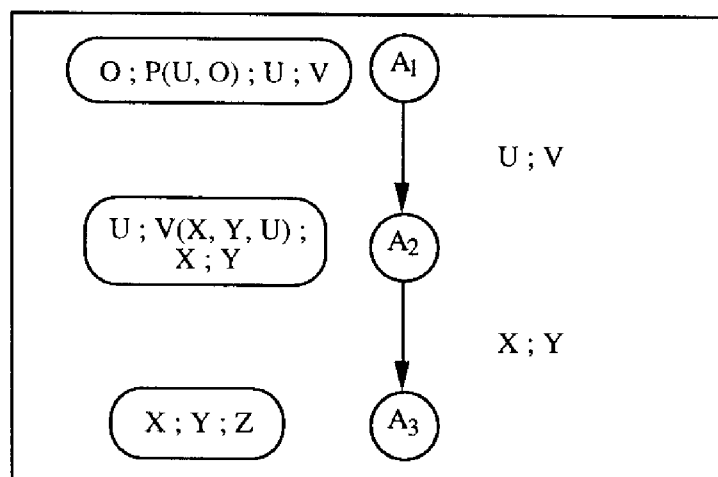


Figure I.16. Répartition des prédicats suivant la structure de la coopération

Soit un graphe de coopération où R est transitive. Chaque agent voit toutes les valeurs exportées par les agents situés au dessous de lui dans le graphe. Par contre, les prédicats évalués par les antécédents de A_j ne peuvent pas être connus par A_j . La connaissance des prédicats est donc directement liée à la structure du groupe. Dans le cas général, R peut ne pas être triviale en terme de prédicats et peut entraîner des relations de dépendances entre des valeurs de prédicats. A titre d'exemple, la figure I.16 donne un cas de dépendances de prédicats. Les données encadrées représentent les

prédicats locaux ou importés connus de chaque agent. Ils donnent la connaissance de chaque agent dans la coopération. Les prédicats sur les flèches sont les prédicats exportés par un agent. Dans le cas de la figure I.16, le prédicat V appartenant à A_2 dépend à la fois des prédicats exportés X et Y , mais aussi du prédicat U local à l'agent A_2 . De plus, comme A_3 n'a pas accès aux prédicats exportés par A_2 , il ne peut connaître ni U , ni V . Par contre, les prédicats U et V sont exportés tels quels vers A_1 , sans porter à sa connaissance directe les prédicats dont U dépend. Ainsi, A_3 connaît X , Y et Z . A_2 connaît U , X , Y et V , et A_1 connaît O , P , U et V .

L'allocation des prédicats aux agents est donc une étape majeure de la conception. Dans le cas particulier où l'on désire exporter une valeur importée, il suffit d'affecter la valeur reçue à un nouveau prédicat dont la valeur sera liée à celle de l'ancien prédicat par la relation d'égalité.

2.3.2.3. Domaines de coopération

Dans le cas général, la relation R est quelconque. Il s'ensuit que, pris dans leur ensemble, tous les agents appartiennent à un système qui ne suit qu'une seule règle : la réflexivité. En fait, intuitivement, les systèmes complexes ne sont pas conçus ainsi, c'est-à-dire ne sont pas conçus globalement, et ne forment pas un tout insécable. Les systèmes complexes sont le plus souvent organisés de façon modulaire, sont structurés et sont découpés en sous-systèmes relativement indépendants entre eux. Cette approche est également utilisée pour regrouper certains agents suivant leurs relations, suivant leurs rôles et suivant les travaux qu'ils doivent accomplir en commun au sein du groupe coopératif. Le découpage de la tâche principale en sous-tâches revient à considérer la relation globale R comme un ensemble de sous-relations s'appliquant sur des sous-ensembles d'agents. Ces sous-relations et leurs ensembles d'agents associés sont appelés des *domaines de coopération*.

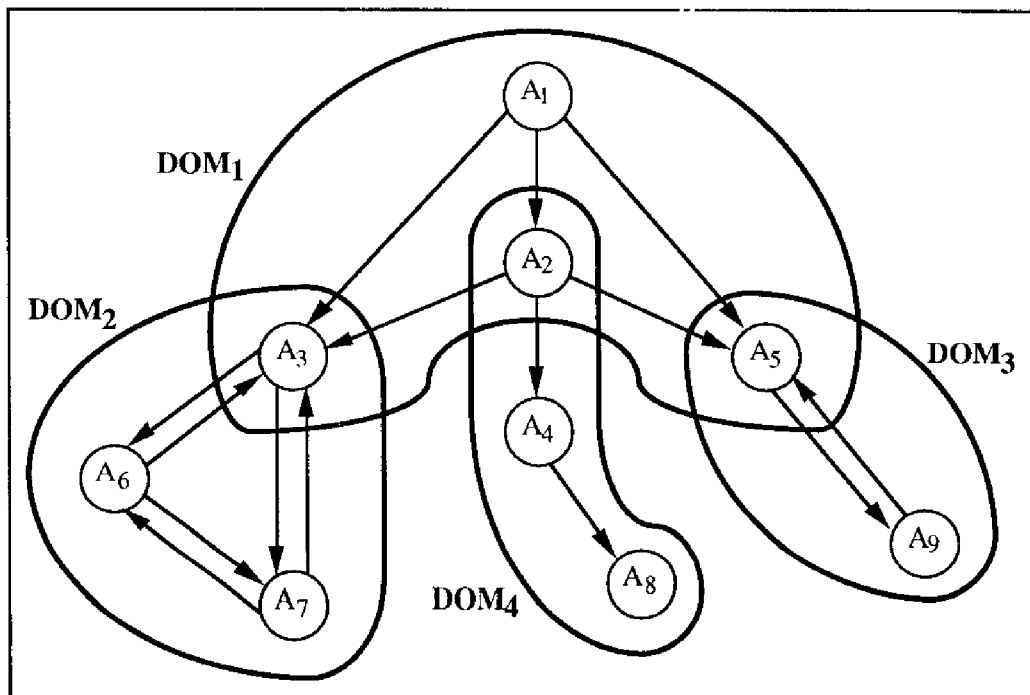


Figure I.17. Découpage du graphe de coopération en plusieurs domaines

A titre d'exemple, le groupe coopératif de la figure I.17 peut être vu de différentes façons. Lorsque l'on regarde A_1 , A_2 , A_3 et A_5 ainsi que les arcs les connectant, ces agents sont en coopération hiérarchique : la coopération R_1 réduite aux sommets A_1 , A_2 , A_3 et A_5 et aux cinq arcs qui les relient, est une restriction transitive de R qui décrit

le domaine DOM_1 . La relation R_2 , restriction de R aux sommets $\{A_3, A_6, A_7\}$ forme le domaine de coopération totale DOM_2 . De même, la relation R_3 , restriction de R aux sommets $\{A_5, A_9\}$ forme un autre domaine de coopération totale, DOM_3 . Finalement, R réduite au sommets $\{A_2, A_4, A_8\}$ forme la relation R_4 qui décrit le comportement du domaine DOM_4 . L'agent A_3 est donc impliqué dans deux sous-coopérations différentes, une hiérarchique, R_1 , l'autre totale, R_2 . Cependant, A_3 fait aussi partie de la relation de coopération globale R , incluant tous les agents, et définie par l'ensemble du graphe de la figure I.17.

Le découpage en domaines permet de diviser la tâche globale associée à la coopération en un ensemble de sous-tâches pouvant s'exécuter de manière autonome ou semi-autonome par rapport aux autres sous-tâches coopératives. R peut être divisée en différents domaines de coopération dédiés à des sous-problèmes, ou sous-applications, cohérents. Ces domaines définissent dans un certain sens des ensembles gérables d'agents.

C'est aussi un deuxième moyen de structurer la coopération, de grouper et de répartir l'information entre les différents domaines [HEW91]. En effet, certaines informations pourront être regroupées à l'intérieur d'un domaine, et de ce fait n'être accessibles qu'aux membres de ce domaine. Il s'agit de la notion d'information privée.

De plus, comme un domaine définit un groupe devant réaliser un travail de manière coopérative, il est nécessaire que les données demeurent cohérentes à l'intérieur de ce même domaine. Cette condition est garantie si chaque prédicat du domaine est affecté de manière unique à un et un seul agent du domaine, qui devient propriétaire de ce prédicat. Seul l'agent propriétaire peut modifier sa valeur, les autres agents du domaine en relation avec le propriétaire se contentant d'accéder à cette valeur.

2.3.2.4. *Dynamicité du graphe de coopération dans le temps*

Le modèle conceptuel précédent ne prend pas en compte l'évolution possible de la structure du groupe coopératif dans le temps. A un instant donné, tous les agents du groupe coopératif n'ont pas besoin d'être présents pour commencer ou pour accomplir le travail coopératif. Pour offrir une telle possibilité, le modèle a été étendu pour définir quels agents peuvent ou doivent être présents ensemble pour initialiser ou effectuer le travail coopératif [DIA93c] [DIA93d].

Le graphe de domaine de coopération, appelé plus simplement graphe de domaine, est défini comme le sous-graphe du graphe de coopération ne comprenant que les agents qui font partie du domaine considéré. Si l'on se réfère à l'exemple de la figure I.17, le graphe de domaine DOM_1 est formé des agents A_1, A_2, A_3 et A_5 , le graphe de domaine DOM_2 est formé des agents A_3, A_6 et A_7 , celui de DOM_3 contient A_5 et A_9 , celui de DOM_4 est formé de A_2, A_4 et A_8 .

Le graphe de domaine de coopération représente les relations entre tous les agents d'un domaine sans tenir compte de l'évolution de la coopération dans le temps. Or, à chaque instant, dans un domaine de coopération, il n'est pas nécessaire que les agents participent tous en même temps au travail associé au domaine.

Pour chacun des domaines de coopération, un graphe instantané de domaine de coopération, appelé plus simplement graphe instantané de domaine, est défini comme un sous-graphe du graphe de domaine : il ne représente que les agents qui coopèrent effectivement à un instant donné dans le domaine considéré.

Parmi tous les sous-graphes des graphes de domaine, certains d'entre eux ne représentent rien ou n'ont aucun sens pour l'application coopérative réalisée. Les sous-ensembles d'agents qui ont une signification pour la réalisation du travail coopératif associé au groupe doivent donc être définis par la sémantique de l'application. Parmi tous les sous-graphes des graphes de domaines, l'application choisit alors ceux qui sont sémantiquement valides donc possibles. Les sous-graphes retenus par l'application sont appelés graphes instantanés valides. La figure I.18 montre deux exemples de sous-

graphes valides du domaine DOM_1 qui pourraient provenir de la structure coopérative de la figure I.17.

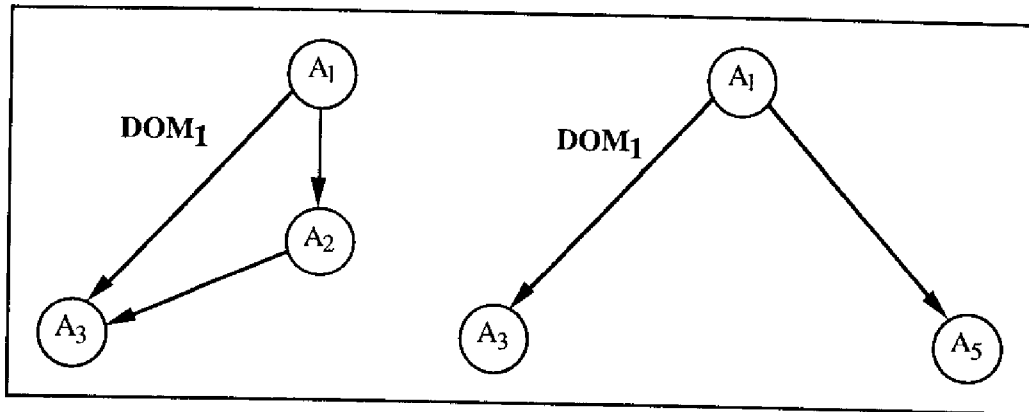


Figure I.18. Exemple de deux graphes instantanés valides provenant du domaine DOM_1 du groupe coopératif de la figure I.15.

Pour définir quels sous-graphes sont valides, l'application doit donner un ensemble de propriétés exprimées par des règles ou des prédicats que ces graphes doivent respecter. Ces règles dépendent bien entendu du travail coopératif à accomplir. A titre d'exemple, quelques critères peuvent être donnés :

Pour un domaine, il faut :

- que tout graphe instantané valide soit connexe.
 - au moins deux agents dans un graphe instantané valide.
 - au moins N agents dans le graphe.
 - au plus N agents dans le graphe.
 - exactement N agents dans le graphe.
 - que les antécédents d'un nœud (par rapport à une racine) soient dans le graphe.
- etc...

Dans un cas plus général, s'il est impossible de trouver de simples critères pour la sélection des graphes instantanés valides, ces graphes peuvent être énumérés explicitement.

Remarque :

L'ajout des configurations valides dans le modèle ne peut s'appliquer que dans le cas de coopérations synchrones. En effet, cette extension cherche à définir les configurations possibles pour lesquelles les coopérants peuvent travailler en même temps. Ceci n'est possible que si les agents peuvent être co-présents en coopération, ce qui correspond à la définition de la coopération synchrone. La coopération asynchrone se déroule sans la co-présence des différents membres : on n'a pas de notion de travail en même temps, et donc on ne peut avoir de configuration instantanée valide.

A priori, l'introduction des graphes instantanés valides semble restreindre la généralité du modèle de coopération. Ce choix a cependant été fait pour plusieurs raisons : tout d'abord, le but recherché est de formaliser et de modéliser la coopération synchrone. En effet, la plupart des modèles proposés s'appliquent davantage aux cas asynchrones, et beaucoup moins aux coopérations de type synchrone. La deuxième raison est de pouvoir, par la suite, utiliser ce modèle dans le cadre de manipulations de données vidéo "temps-réel" multimédias. Or, ces données n'interviennent que dans le cadre de coopérations synchrones.

3. Exemples d'applications multimédias

Les parties précédentes ayant introduit la double problématique associée à la conception d'applications distribuées multimédias coopératives, cette dernière partie de ce premier chapitre énumère quelques applications ou domaines d'applications multimédias qui existent aujourd'hui. Cette énumération ne se veut en aucun cas exhaustive ; elle a juste pour objectif d'illustrer les propos précédents par des cas concrets, et ce en balayant le plus largement possible le champ des familles d'applications multimédias qu'il est possible de rencontrer.

3.1. Les applications multimédias locales

La première famille d'applications multimédias présentée est un peu en marge des travaux exposés dans cette thèse puisque ces applications ne sont pas distribuées, et s'exécutent sur une seule machine. Elles sont apparues dès que les traitements de données multimédias (audio, vidéo en plus du texte et du graphique) ont été intégrés sur les ordinateurs et dès que les supports de stockage de masse - en particulier le CD-ROM - ont permis de sauvegarder et de diffuser les quantités de données induites par ces applications.

Parmi ces applications, les domaines qui ont le plus utilisé ces technologies sont le domaine de l'enseignement assisté par ordinateur (EAO) et le domaine de la documentation. Ainsi, de très nombreux systèmes d'EAO multimédias sont apparus comme SteelDEM [AMI93], TRAIN [AUG93] ou VACBI, le système de formation professionnel d'Airbus pour les pilotes et agents de maintenance, sur lequel nous reviendrons au chapitre 5.

Le domaine de la documentation a également massivement utilisé les technologies multimédias avec en particulier toutes les encyclopédies multimédias. Il faut noter toutefois que toutes ces applications présentent des documents hypermédias dont les hyperliens permettent de naviguer dans le document.

3.2. Les applications multimédias distribuées

Les autres applications multimédias, quant à elles, utilisent des supports de communications distants et s'exécutent sur plusieurs machines : ce sont les applications distribuées multimédias. Parmi ces applications (appelées aussi *services*), le CCITT distingue deux catégories [CCI88] :

- les services interactifs, incluant les services conversationnels (audioconférence et visioconférence), les services de messagerie (courriers électroniques multimédias) et les services de consultation (bases de données multimédias) ;
- les services de distribution, sans contrôle de présentation de la part de l'utilisateur (télévision par câble), ou avec contrôle (jeux électroniques coopératifs).

Dans ce contexte, il est possible de distinguer deux types d'applications : les applications *différées* et les applications *temps réel*.

Les applications multimédias *différées* impliquent le rapatriement et le stockage de documents multimédias en provenance de site(s) distant(s) ; la présentation de ces documents est effectuée dans un second temps. Dans ce contexte, la principale contrainte que peut subir l'utilisateur est la durée séparant sa requête de la présentation finale ; elle est liée au temps de transfert du document. Une fois que le document multimédia est disponible localement dans son intégralité, l'application est assimilable à une application locale ; en cela, ces applications présentent un caractère asynchrone.

Les applications multimédias *temps réel* impliquent également le transfert de documents multimédias, mais la restitution de ces documents se fait en temps réel, c'est-à-dire au fur et à mesure de leur acquisition au niveau de la source et/ou du réseau.

En cela, ces applications présentent un caractère synchrone, destiné à améliorer l'interactivité entre leur utilisateurs.

La suite présente quelques unes des applications multimédias les plus illustratives et en dégage les caractéristiques majeures.

Les bases de données multimédias distribuées

Les bases de données multimédias distribuées permettent d'accéder à des quantités d'informations beaucoup plus importantes que les bases de données traditionnelles. Elles possèdent les fonctionnalités de base (classement, recherche par critères, etc.), mais tiennent compte du fait que les données sont à présent multimédias. L'intérêt d'une base de données multimédias apparaît dans des domaines comme le médical [KAR90], les guides touristiques ou les pages jaunes en permettant des consultations plus conviviales. Les besoins de ce type d'applications diffèrent selon le degré d'interactivité entre l'utilisateur et la base. Généralement, le temps de réponse de la base n'est pas le critère de qualité le plus significatif, et ce type d'applications ne présente pas de fortes contraintes de temps : ce sont le plus souvent des applications différées. Actuellement, le système de base de données le plus en vogue est le World Wide Web permettant à ses utilisateurs de consulter des documents multimédias par l'intermédiaire de logiciels tels que *Mosaic* ou *Netscape*.

Les messageries multimédias

Le concept de messagerie permettant l'échange de documents électroniques composés exclusivement de caractères a été étendu pour permettre l'échange asynchrone (ou différé) de documents multimédias entre plusieurs utilisateurs. Les documents échangés ont ainsi été enrichis et peuvent contenir non seulement du texte, mais aussi du son, de la vidéo ou tout autre support d'information permettant la transmission de messages plus riches. D'autres part, des spécifications sur les relations entre les différents éléments peuvent être précisées. Cette dernière particularité offre la possibilité de composer des messages hypertextes et hypermédias. MIME¹¹ [BOR92] et MMMS¹² [BER93] sont des exemples de messageries multimédias asynchrones.

Il existe également des messageries qui respectent des contraintes temps réel ; c'est le cas du *talk* ou d'*IRC* (Internet Relay Chat) qui permettent à deux ou un groupe d'utilisateurs de dialoguer en temps réel. De plus, un système comme *CHAOS* (Commitment Handling Active Office System) [BIG89] intègre des notions de coopération entre les interlocuteurs ; par exemple, ce système classe, répertorie et archive des connaissances pour supporter des communications humaines. Les messages manipulés sont semi-structurés de manière à ce que le système puisse les interpréter. Dans un groupe, la connaissance commune est constituée par l'évolution du groupe et par son historique. Cette connaissance commune permet d'informer les nouveaux arrivants. Ces systèmes peuvent être vus comme les précurseurs textuels des systèmes d'audioconférence et de visioconférence.

Les éditeurs multi-utilisateurs

Dans les applications d'édition multi-utilisateurs, un groupe d'utilisateurs est autorisé à manipuler et à modifier un même document en temps réel. Deux aspects majeurs caractérisent ces applications : le partage et le contrôle d'accès aux informations (comme dans *XTV*[CHU94]). *GROVE* (GROup Outline Viewing Editor) [ELL91], *DistEdit* [KNI90], *GroupDesign* [KAR93] et *MBLINK* [3AR85] - pour ne citer que ces

¹¹MIME : Multipurpose Internet Mail Extension. MIME est une extension du mail électronique classique SMTP (Simple Mail Transfert Protocol) [POS82] pour Internet.

¹²MMMS : MultiMedia Mail Service. Ce service de messagerie repose sur la recommandation X400 du CCITT.

quatre - sont des éditeurs multi-utilisateurs ; les deux premiers sont des éditeurs textuels ; les deux derniers sont des éditeurs graphiques et/ou multimédias.

Les systèmes de télé-enseignement, de télétravail ...

Le principe d'une application de télé-enseignement est de permettre à un enseignant de délivrer son cours en temps réel (et sans se déplacer) à plusieurs élèves répartis sur des sites distants. Des applications de ce style ont vu le jour (au moins à l'état de prototypes) comme CO-LEARN [DER93] [CRO94], télé-amphi [GUI92] ou encore VACBI étendu [OWE95b]. Toutefois, les applications et outils développés pour le télé-enseignement ouvrent aussi la voie à la téléformation professionnelle et au télétravail en règle générale.

3.3. Le cas particulier de la visioconférence

Un type important d'applications distribuées multimédias est formé par la famille des applications de visioconférence. Cette famille revêt une importance de tout premier plan dans cette thèse puisque la conception de ce type d'applications est l'objectif même des travaux présentés ici. Aussi, cette partie va présenter certains systèmes de visioconférence en les confrontant à la problématique exposée dans ce chapitre ; cette partie ne peut être exhaustive, le nombre des systèmes existants s'élevant à plusieurs centaines d'unités. Toutefois, les systèmes évoqués seront classés en trois groupes, en fonction du type de réseau qu'ils utilisent : (a) les systèmes qui fonctionnent sur réseaux locaux, ces réseaux offrant une bande passante assez importante, mais peu ou pas de garanties temporelles, (b) ceux qui fonctionnent sur le RNIS, ce réseau offrant des garanties temporelles fortes mais une bande passante réduite, et enfin, (c) les derniers qui fonctionnent sur le réseau Internet, ce réseau n'offrant qu'une faible bande passante et aucune garantie temporelle.

Applications de visioconférences sur réseaux locaux

Les premiers systèmes de visioconférence sont apparus sur des réseaux locaux et permettent la visioconférence entre les membres d'une même institution. Par exemple, la société Paradise Software a développé les applications PSVC (Paradise Software VideoConference) [PAR94] et UNIFLIX [PAR94]. Ces applications utilisent les capacités de compression de la carte d'acquisition vidéo Parallax et le format utilisé pour la vidéo est M-JPEG (Moving-JPEG) ; UNIFLIX permet, par rapport à PSVC, une décompression logicielle, celle de PSVC étant forcément réalisée par la carte Parallax. Ces deux outils n'intègrent toutefois pas de mécanismes de synchronisation multimédia et ne permettent pas la visioconférence pour un groupe de plusieurs utilisateurs (UNIFLIX possède juste des possibilités de diffusion, mais sans contrôle temporel et sans contrôle des participants).

TSVS [DIA94a] est aussi un système de visioconférence point à point qui utilise le format de compression M-JPEG de la carte Parallax. A cause du débit requis par ce format de compression, cette application ne peut pas utiliser les réseaux longue distance actuels, sauf en dégradant fortement la qualité vidéo. Par contre, cette application intègre des mécanismes de synchronisation (déjà présentés dans la première partie de ce chapitre) mais dont l'efficacité requiert certaines conditions favorables de la plate-forme utilisée. Il faut noter aussi, qu'une version coopérative de TSVS existe ; elle sera présentée au chapitre 4.

De même, l'application de visioconférence PNSVS présentée dans les chapitres 2 et 3 a été développée sur réseaux locaux. Ce système a les mêmes caractéristiques que TSVS, excepté que la synchronisation multimédia est implémentée en utilisant un schéma modélisé par un TSPN.

Applications de visioconférence au dessus du RNIS

Les applications de visioconférence qui utilisent le RNIS permettent des communications entre des interlocuteurs distants de plusieurs centaines, voire milliers de kilomètres. Une version de TSVS (utilisant une taille d'image réduite) a été développée au dessus du RNIS. Toutefois, le développement d'applications au dessus de ce réseau reste le domaine des industriels. C'est le cas pour ProShare 200 d'Intel (diffusé en France grâce au concours de France Télécom EGT) qui est un exemple parmi tant d'autres (tous ces outils ont quasiment les mêmes fonctionnalités). Ce logiciel utilise pour la vidéo le format compressé H320, et respecte la normalisation T120 pour toutes les interactions entre interlocuteurs (travail coopératif). Ce système de visioconférence n'intègre pas de mécanisme de synchronisation multimédia, mais le réseau RNIS étant synchrone, seules les perturbations au niveau du système opératoire et des cartes de présentations peuvent dégrader les propriétés de synchronisation entre les flux ; ces perturbations sont par ailleurs très réduites lorsque ProShare 200 fonctionne seul. ProShare 200 se restreint toutefois à des communications point à point.

Applications de visioconférence sur Internet

De nombreux systèmes de visioconférence expérimentaux ont été développés au dessus du réseau Internet. Ils utilisent les protocoles transport UDP¹³ [POS80] ou RTP¹⁴. RTP présente, par rapport à UDP qui offre le service transport minimal, des mécanismes permettant de prendre en compte les contraintes temps réel des applications. Ces mécanismes utilisent comme référence une estampille transportée dans chacun des paquets RTP, et qui représente la contrainte temps-réel à respecter. Toutefois, comme nous l'avons déjà vu précédemment, les estampilles temporelles n'offrent qu'un pouvoir sémantique réduit ; de plus, les mécanismes du protocole RTP, qui reposent sur le service IP asynchrone, ne permettent pas de garantir les contraintes temps réel de l'application, excepté si des réservations sont faites au niveau de la bande passante des lignes et des capacités de traitement des noeuds de commutation, avec un protocole comme RSVP¹⁵. Si les ressources nécessaires à une application peuvent être réservées, alors l'asynchronisme du service et du protocole IP sera réduit, voire annulé, et le protocole RTP pourra garantir un service "temps-réel".

Les systèmes de visioconférence sur Internet utilisent des algorithmes de compression qui engendrent des débits très faibles en sortie avec des qualités vidéo assez faibles en général. De plus, ces systèmes n'intègrent la plupart du temps pas de mécanisme de synchronisation multimédia, leur soucis principal étant l'algorithme de compression vidéo. La suite présente quelques unes de ces applications.

CU-SeeMe

CU-SeeMe [DOR95] a été conçu pour les plates-formes peu puissantes. Ce système utilise son propre algorithme de compression (non normalisé). Il ne permet pas non plus les communications multipoints, sauf au travers de réflecteurs, mais cette technique ne s'avère pas vraiment efficace.

NV

NV (Network Video) [FRE94] intègre plusieurs codecs comme JPEG, CellB et l'algorithme propriétaire non normalisé NV. Il n'intègre pas de mécanismes de synchronisation, ni même des mécanismes de coopération.

¹³UDP : User Datagram Protocol.

¹⁴RTP : Real time Transport Protocol [SCH96b].

¹⁵RSVP : ReSerVation Protocol.

VIC

VIC (Video Conference) [MCC94] intègre également plusieurs codecs vidéo : H261, NV, CellB et JPEG. Il n'intègre par contre pas de mécanismes de synchronisation et de coopération.

IVS

IVS (INRIA Visioconference system) [TUR93] [TUR95] utilise l'algorithme de compression H261. Il utilise également des mécanismes de diffusion (IP étendu pour le multicast [DEE88]), et intègre des protocoles de contrôle de transmission, de contrôle de débit et de contrôle de congestions. Rendez-Vous, son successeur, intègre, en plus, des mécanismes de synchronisation multimédia basés sur RTP version 2 et un ordonnancement spécifique des traitements sur les flux audio et vidéo de l'application.

COMMUNIQUE

COMMUNIQUE est un logiciel de visioconférence développé par la société Insoft ; Ce logiciel utilise plusieurs algorithmes de compression (H261, JPEG, CellB) et peut fonctionner sur plusieurs plates-formes de communication (réseau locaux, RNIS et Internet). Il intègre également des mécanismes de diffusion et des mécanismes de synchronisation multimédia ; toutefois, la synchronisation est obtenue en augmentant le temps de stockage des données audio et vidéo reçues ce qui est donc extrêmement préjudiciable à l'interactivité entre les interlocuteurs, et qui, de toute manière, ne garantit pas la synchronisation puisque, le support étant asynchrone, la gigue n'est pas bornée ; on peut donc avec ce système dégrader l'interactivité sans obtenir le niveau de synchronisation souhaité.

Toutefois, COMMUNIQUE, de même qu'UNIFLIX ne sont pas seulement des logiciels de visioconférence ; ils intègrent également des outils de tableau blanc partagé et de partage d'applications.

Conclusion

L'objectif de ce chapitre a été de poser la double problématique associée à la conception d'applications distribuées multimédias coopératives. Les deux problèmes à résoudre sont donc d'une part la manipulation des flux de données multimédias en permettant le plus haut niveau de qualité de service possible, ou du moins celui demandé par les utilisateurs, et d'autre part la gestion des participants à l'activité coopérative (avec entre autres les problèmes de communication dans un groupe coopératif). Les problèmes ont été posés et des modèles de solution ont déjà été proposés.

De plus, l'état de l'art qui a terminé ce chapitre a montré que les applications actuelles et en particulier les systèmes de visioconférence existants n'intègrent pas ces notions de coopération et de synchronisation multimédia, et lorsqu'ils les prennent en compte, ils les implémentent par des méthodes ad-hoc utilisables uniquement dans un cas précis et ne garantissant pas les contraintes à respecter.

La suite de cette thèse va donc présenter nos travaux dans le domaine de la conception et du développement de systèmes de visioconférence intégrant des mécanismes de gestion de la synchronisation multimédia, de la qualité de service et de la coopération, et ce en utilisant des formalismes qui pourront être généralisés à toutes les applications multimédias coopératives. Nous ne nous soucierons pas des problèmes liés aux algorithmes de compression vidéo, comme c'est souvent le cas dans les travaux qui visent à concevoir des applications pour le réseau Internet actuel ; nous

considérons que nous avons à notre disposition un support de communication haut-débit, par exemple un réseau ATM¹⁶.

¹⁶ATM : Asynchronous Transfert Mode.

Chapitre II

Gestion de la synchronisation multimédia et de la qualité de service dans une application de visioconférence

Introduction

Ce chapitre décrit l'application de visioconférence réalisée : PNSVS ¹. Le chapitre 1 a montré que les deux points essentiels à résoudre pour la conception d'applications distribuées multimédias coopératives sont :

- la gestion de la synchronisation multimédia et de la qualité de service en général ;
- l'introduction de mécanismes de communication de groupe et de coopération.

L'objectif de ce chapitre est donc, dans un premier temps, de montrer comment il est possible de réaliser une application de visioconférence point à point synchronisée dont la qualité de service est négociée, et ce en utilisant un système Unix asynchrone, et un protocole de communication "Best Effort" - tel UDP - au dessus d'un réseau asynchrone (Ethernet, FDDI ou ATM). En particulier, et cela n'a jusqu'à aujourd'hui pas été fait à notre connaissance, une modélisation des comportements de l'application prenant en compte les contraintes du système opératoire et les requêtes de l'utilisateur est proposée. Cette modélisation dérive le modèle TSPN - qui représente les besoins de l'utilisateur - en ajoutant les éléments nécessaires à la modélisation des mécanismes permettant d'obtenir la synchronisation requise sur le système opératoire dont nous disposons. Cette modélisation est très importante pour la compréhension des mécanismes de cette application et de la problématique générale.

L'introduction de mécanismes de coopération sera vue plus avant.

A la base, PNSVS est une application de visiophonie permettant à deux utilisateurs terminaux de communiquer par l'intermédiaire de la voix et de l'image, de façon interactive et en temps réel. Cette application consiste en deux flux audio et vidéo capturés sur une station émettrice, transmis sur un réseau de communication, puis reçus

¹PNSVS : Petri Net Synchronized Visioconference System.

et présentés à l'utilisateur final sur la station réceptrice, et ce dans les deux directions (figure II.1).

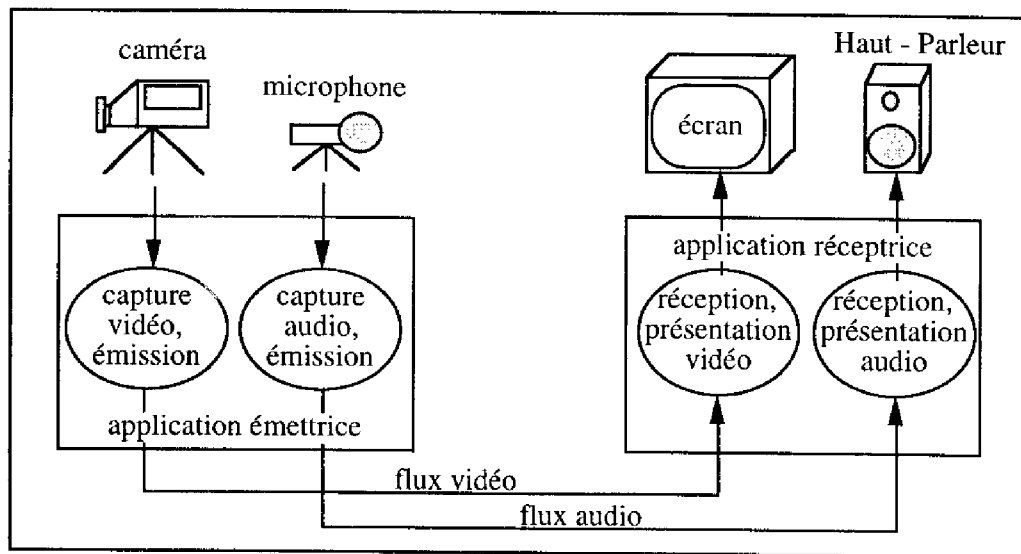


Figure II.1. Principe de la visioconférence (mono-directionnelle)

Les paramètres de qualité de service d'une application de visioconférence se divisent en deux groupes : les paramètres statiques et les paramètres dynamiques. Les paramètres statiques, comme la taille d'une image, sa qualité, ou la qualité du flux audio sont importants car ils interviennent directement sur les ressources qui seront nécessaires à l'application pour fonctionner correctement. Par contre, ils n'ont aucune influence sur les algorithmes des différents mécanismes de cette même application. Les paramètres dynamiques, en plus d'intervenir sur la quantité des ressources à utiliser, interviennent aussi directement sur le code de l'application et nécessitent d'intégrer à cette application des mécanismes spécifiques pour les respecter. Ces paramètres dynamiques qui vont donc être le leitmotiv de tous les mécanismes décrits dans la suite de ce chapitre appartiennent à trois familles distinctes : les paramètres de qualité de la synchronisation multimédia, les paramètres dynamiques de qualité de présentation, et le retard de bout en bout.

- Les paramètres de qualité de la synchronisation multimédia définissent les critères de synchronisation intra et inter-flux. Ces critères sont d'ordre temporels : pour la synchronisation intra-flux, ils définissent les giges maximales acceptables sur les images ou sur les données audio. Pour la synchronisation inter-flux, ils définissent le décalage temporel maximal acceptable entre les flux audio et vidéo ; ce décalage appelé aussi dérive inter-flux est le résultat des effets cumulés des giges qui apparaissent sur la présentation des objets des différents flux. Les paramètres de synchronisation pour PNSVS sont modélisés par un TSPN, comme par exemple celui de la figure I.9.

Il faut donc intégrer dans l'application de visioconférence des mécanismes de synchronisation permettant de respecter tous les critères de qualité de synchronisation intra et inter-flux

- Les paramètres de qualité de présentation sont essentiellement des paramètres statiques n'ayant pas d'influence sur les algorithmes de l'application. Toutefois, l'un d'entre eux est dynamique ; dans ce type d'application lorsqu'une donnée est perdue elle entraîne une perte importante de synchronisation. Pour éviter ce problème, il faut substituer à la donnée perdue une donnée équivalente temporairement (généralement la donnée précédente). Ceci conduit cependant à une discontinuité dans la présentation. Il faut donc faire en sorte de réduire au maximum le nombre de discontinuités, chaque discontinuité nuisant à la qualité générale de la présentation.

- Le retard de bout en bout définit le délai maximum acceptable entre les deux interlocuteurs. Ce délai correspond au retard entre le moment où l'utilisateur émetteur prononce un mot, et le moment où l'utilisateur récepteur reçoit le son associé à ce mot. Ce retard ne doit pas être supérieur à une certaine limite pour ne pas nuire au niveau d'interactivité du dialogue entre les deux interlocuteurs, et l'application de visioconférence devra donc le contrôler.

Tous ces paramètres dynamiques de qualité de service devront être pris en compte par l'application qui devra donc intégrer des mécanismes spécifiques permettant de les respecter. Aussi, ce chapitre se divise en 4 parties :

- la première présente une modélisation des mécanismes de gestion de la qualité de service dans PNSVS. Pour cela, cette partie modélise d'abord les mécanismes de synchronisation multimédia, puis ceux qui permettent de réduire le nombre de discontinuités et de contrôler le retard de présentation de bout en bout ;
- la seconde montre comment les mécanismes de synchronisation multimédia ont été implémentés sur une plate-forme composée de station Sun Solaris 2 ;
- la troisième partie montre comment la qualité de service est négociée et renégociée dans PNSVS 1 ;
- et enfin, la quatrième propose une série de mesures destinées à montrer l'efficacité des mécanismes proposés.

1. Modélisation des mécanismes de gestion de la qualité de service

L'objectif de cette première partie consiste à mettre en évidence les conséquences des problèmes induits par l'utilisation d'un support asynchrone, et à modéliser les mécanismes de synchronisation dérivés qui permettent de respecter les contraintes temporelles exprimées par l'utilisateur et représentées par un TSPN. Toutefois, pour illustrer ces principes, les exemples que nous prenons se rapportent à une visioconférence dont les contraintes de synchronisation sont exprimées par un TSPN de la forme de celui de la figure I.9 (ou I.10), c'est-à-dire une application qui doit présenter deux flux audio et vidéo en parallèle avec des contraintes de gigue et de dérive inter-flux variables.

1.1. Mécanismes de synchronisation multimédia

Ainsi, cette partie va tout d'abord prendre deux exemples pour illustrer un comportement caractéristique des systèmes asynchrones et va proposer des mécanismes permettant de respecter les contraintes de synchronisation représentées sur la figure I.9 ; ces mécanismes seront d'abord modélisés dans le cadre du premier exemple, puis le modèle général sera proposé.

1.1.1. Latence des cartes multimédias et décalage des synchronisations inter-flux

Soit le TSPN de la figure I.9. Ce TSPN de présentation montre comment les objets du scénario audio et vidéo doivent être synchronisés. En particulier pour la synchronisation inter-flux, aux problèmes de dérive près, le son i doit être synchronisé à l'image i . Toutefois, comme cela a déjà été évoqué précédemment, les cartes de présentation multimédia n'ont pas toutes les mêmes temps de latence. Ainsi, si l'on considère une carte vidéo ayant un temps de latence de 50 ms et une carte audio dont le temps de latence est 250 ms, alors si le moteur de l'application respecte le TSPN de présentation de la figure I.9, la présentation finale ne sera pas synchronisée. En effet, la partie son sera retardée de 200 ms par rapport à la partie vidéo, i.e. le son i ($i \in N$) sera

synchronisé avec l'image $i+2$, même si le moteur de l'application a synchronisé au départ l'image i avec le son i .

Pour résoudre ces problèmes de temps de latence différents sur les différentes cartes, des décalages doivent être introduits dans les synchronisations inter-flux (on parle aussi de rendez-vous décalés, si on apparente une synchronisation inter-flux à un rendez-vous [OWE95c]). En effet, la différence entre les temps de latence de la carte son et de la carte vidéo étant de 200 ms, il suffit de synchroniser au niveau du moteur de synchronisation le son i avec l'image $i-2$, ce qui - après passage dans les cartes de présentation multimédia - donnera une présentation dans laquelle le son i et l'image i seront synchronisés. Sur cet exemple, le TSPN modélisant le moteur de synchronisation de l'application, avec les rendez-vous décalés, est représenté sur la figure II.2 :

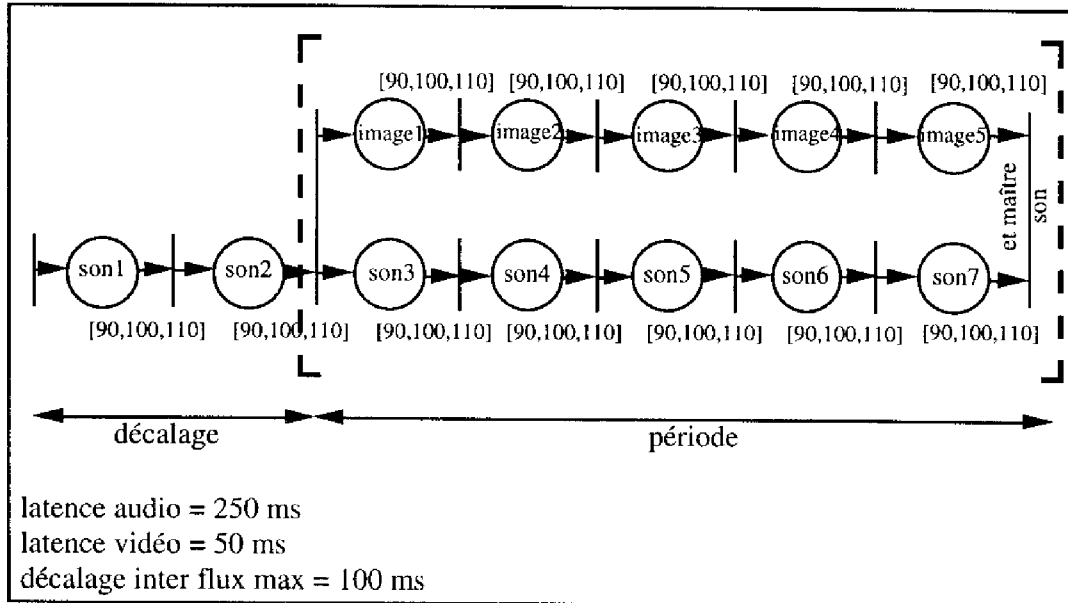


Figure II.2. TSPN applicatif tenant compte des temps de latence matériels

Cependant, l'exemple choisi est idéal, car la différence entre les temps de latence audio et vidéo est multiple du temps nominal de présentation d'un objet. Si ce n'est pas le cas, avec par exemple des temps de latence audio et vidéo égaux respectivement à 230 et 50 ms, alors la différence entre les temps de latence est de 180 ms. Dans ce cas là, comme dans le précédent, le décalage qui doit intervenir dans les rendez-vous entre les flux audio et vidéo est également de deux objets représentant 200 ms de décalage. Cependant, après ce décalage de 200 ms (2 objets), il subsiste un décalage résiduel de 20 ms entre les flux audio et vidéo. Aussi, pour que la dérive inter-flux soit inférieure à 100 ms (cf. exemple de la figure I.9), la période de synchronisation inter-flux du TSPN doit être changée. Si la synchronisation inter-flux était faite toutes les 5 images, à cause des 20 ms de décalage qui n'ont pu être éliminées, la dérive inter-flux par rapport au flux audio serait comprise dans l'intervalle $[-80 \text{ ms}, 120 \text{ ms}]$, ce qui en valeur absolue peut être supérieur aux 100 ms autorisées par la qualité de service. Aussi, une synchronisation inter-flux devra être faite toute les 4 images, ce qui place la dérive inter-flux dans l'intervalle $[-60 \text{ ms}, 100 \text{ ms}]$. Le TSPN applicatif, moteur de la synchronisation applicative, ainsi obtenu est représenté sur la figure II.3.

Les TSPN décrivant le scénario de synchronisation multimédia et le comportement de l'application ont donc des formes différentes, à cause des rendez-vous décalés qui permettent de résoudre les problèmes de temps de latence des cartes multimédias. De plus, la période de synchronisation inter-flux peut être modifiée. La partie suivante donne d'ailleurs les formules qui permettent de calculer, en fonction des temps de

latence et de la période du TSPN de présentation, le décalage à introduire au niveau applicatif, ainsi que la nouvelle période de synchronisation.

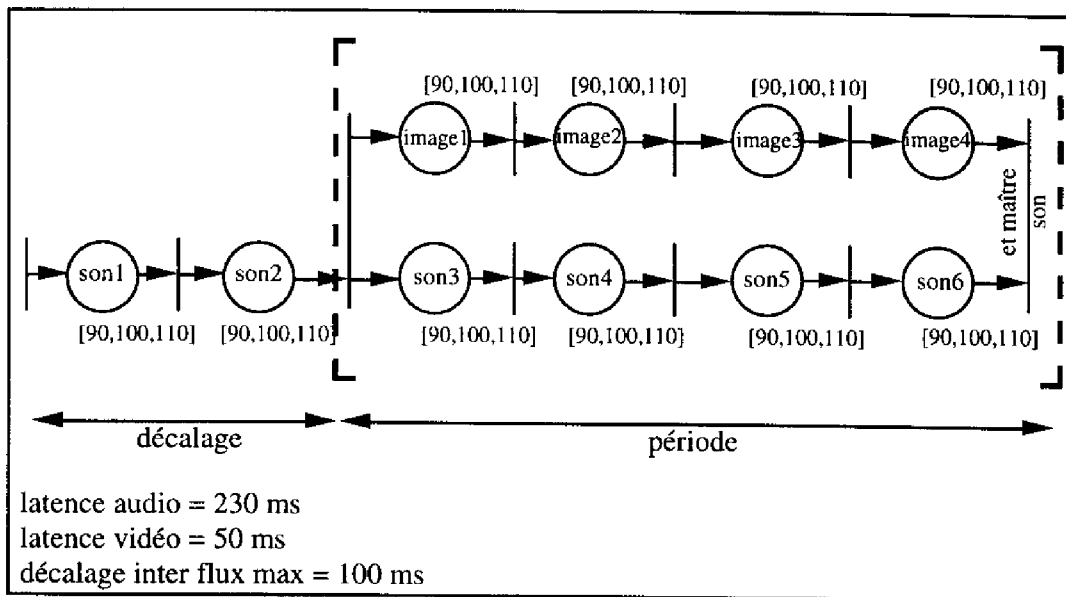


Figure II.3. TSPN applicatif tenant compte des temps de latence matériels et de la période de synchronisation inter-flux

1.1.2. Calcul du décalage des rendez-vous et de la période de synchronisation du TSPN application

- Soit $[t_{\min}, t_{\text{nom}}, t_{\max}]$ les temps de présentation minimal, nominal et maximal associés à une image ;
- Soit $[t'_{\min}, t'_{\text{nom}}, t'_{\max}]$ les temps de présentation minimal, nominal et maximal associés à un son ;
- Soit t_l et t'_l les temps de latence des cartes vidéo et audio ;
- Soit d_{\max} la dérive inter-flux maximale autorisée (en valeur absolue) ;
- Soit N_p le nombre d'images par période de synchronisation inter-flux (on suppose ici que le nombre d'objets audio N'_p sur cette même période de synchronisation est aussi égal à N_p) ;
- Soit N_d le nombre d'objets de décalage permettant de résoudre les problèmes de latences matérielles. N_d est entier. Il est négatif si la présentation des images doit être avancée, et positif si c'est la présentation des sons qui doit être avancée.
- Soit d la dérive entre les flux audio et vidéo. $d < 0$ si le flux audio est présenté avant la vidéo, et $d > 0$ si le flux audio est présenté après la vidéo. d possède deux composantes $d_{\text{décal}}$ et d_p . $d_{\text{décal}}$ est la dérive introduite lors du décalage des rendez-vous (par exemple lorsque la différence des latences n'est pas multiple de t_{nom}). d_p est la dérive inter-flux qui peut apparaître sur une période de synchronisation.

On a :

$$N_d = \text{arrondi} \left(\frac{t'_l - t_l}{t_{\text{nom}}} \right) \quad (1)$$

avec arrondi qui renvoie l'entier le plus proche d'un nombre rationnel.

La dérive liée au décalage est donc :

$$d_{\text{décal}} = (|t'_{i-1} - t_i| - (N_d \times t_{\text{nom}})) \times \frac{t_i - t'_{i-1}}{|t_i - t'_{i-1}|} \quad (2)$$

De plus, la dérive sur une période est comprise entre ces deux cas extrêmes qui consistent à considérer des temps de présentation minimum sur un flux et maximum sur l'autre, soit :

$$N_p \times (t_{\text{min}} - t'_{\text{max}}) \leq d_p \leq N_p \times (t_{\text{max}} - t'_{\text{min}})$$

On a :

$$d = d_{\text{décal}} + d_p \quad (3)$$

soit :

$$N_p \times (t_{\text{min}} - t'_{\text{max}}) + d_{\text{décal}} \leq d \leq N_p \times (t_{\text{max}} - t'_{\text{min}}) + d_{\text{décal}}$$

Or on veut :

$$-d_{\text{max}} \leq d \leq +d_{\text{max}}$$

soit :

$$\begin{cases} N_p \times (t_{\text{max}} - t'_{\text{min}}) + d_{\text{décal}} \leq d_{\text{max}} \\ N_p \times (t_{\text{min}} - t'_{\text{max}}) + d_{\text{décal}} \geq -d_{\text{max}} \end{cases}$$

d'où :

$$\begin{cases} N_p \leq \frac{d_{\text{max}} - d_{\text{décal}}}{t_{\text{max}} - t'_{\text{min}}} \\ N_p \leq \frac{d_{\text{max}} + d_{\text{décal}}}{t'_{\text{max}} - t_{\text{min}}} \end{cases}$$

donc :

$$N_p = E \left(\min \left\{ \frac{d_{\text{max}} - d_{\text{décal}}}{t_{\text{max}} - t'_{\text{min}}}, \frac{d_{\text{max}} + d_{\text{décal}}}{t'_{\text{max}} - t_{\text{min}}} \right\} \right) \quad (4)$$

Ainsi, à partir d'un TSPN du style de celui de la figure I.9 qui représente les contraintes de synchronisation de l'utilisateur, il est possible d'obtenir - en utilisant les formules (1) et (4) du calcul du décalage des rendez-vous et de la période de synchronisation inter-flux - le TSPN modélisant les mécanismes de synchronisation de l'application en environnement asynchrone.

1.2. Réduction du nombre de discontinuités et contrôle du retard de présentation de bout en bout

1.2.1. Principe

Les mécanismes de synchronisation intra-flux peuvent engendrer des discontinuités dans la présentation des flux multimédias. En effet, si un objet est perdu dans le système de communication, ou s'il est retardé, cet objet ne sera pas présenté, et un traitement exceptionnel sera effectué pour palier à cette déficience : un objet audio perdu ou considéré comme tel (s'il est retardé) sera remplacé par un objet audio vide, et

une image perdue sera remplacée par la dernière image reçue. Cependant, chacun de ces traitements exceptionnels (perte ou duplication) induit une rupture dans la séquence de présentation du flux, et induit donc une ou plusieurs discontinuités préjudiciables à la qualité de la présentation. De même, le flux audio étant maître, il peut provoquer des accélérations du flux vidéo, ce qui conduit à la non présentation de certains objets vidéo, et induit donc de nouvelles discontinuités.

Ces discontinuités correspondent à des abaissements de la qualité de la présentation, et leur nombre doit être réduit. D'ailleurs, le nombre maximum de discontinuités acceptables sur chaque flux et par seconde est un paramètre de qualité de service qui permet à l'utilisateur de préciser au système la qualité de service minimale qu'il peut tolérer.

Ces discontinuités correspondent toujours soit à une perte d'un objet par le système de communication, soit au retard d'un de ces objets causé par un problème de gigue positive du système asynchrone. Les problèmes de pertes ne peuvent pas être résolus au niveau de l'application. En revanche, si l'application stocke suffisamment longtemps les données reçues avant de les présenter, alors les problèmes de gigue peuvent être améliorés, car elle pourra absorber des gigues très importantes. Plus l'application stockera les données longtemps, plus la probabilité de présence de l'objet à traiter dans le tampon de stockage sera élevée. Aussi, la solution pour résoudre les problèmes de discontinuité réside dans un stockage au niveau de l'application.

Toutefois, il apparaît rapidement que le stockage ne constitue pas une solution idéale car il introduit des retards de présentation qui sont préjudiciables à l'interactivité du dialogue entre les deux interlocuteurs. L'application ne peut donc pas stocker des données trop longtemps sous peine de dépasser le retard de présentation de bout en bout autorisé. De ce fait, et c'est le point important, le retard de bout en bout constitue le paramètre de qualité de service qui va permettre de contrôler le niveau d'interactivité entre les deux interlocuteurs, et donc imposer la taille des tampons de réception associés à chacun des flux audio et vidéo [JEF94a].

Le retard de bout en bout effectif est créé à différents endroits du système distribué, i.e. au niveau du système de communication, du système opératoire, de l'application, et des cartes multimédias. En fait, ce retard se décompose en :

- un temps de latence des cartes multimédias en acquisition (t_{le} et t'_{le} pour respectivement les cartes vidéo et audio) ;
- un temps de traitement en émission ($t(\text{trait_e})$ pour le traitement vidéo et $t'(\text{trait_e})$ pour le traitement audio) ;
- un délai de communication d_{com} du système de communication ;
- un temps de stockage t_{stock} dans les tampons de l'application ;
- un temps de traitement au niveau de l'application réceptrice ($t(\text{trait_r})$ et $t'(\text{trait_r})$ pour les traitements vidéo et audio) ;
- et un temps de latence en présentation des cartes multimédias (t_{lr} et t'_{lr} pour les cartes vidéo et audio).

Soit R_{max} le retard de présentation maximal autorisé. Pour respecter cette borne, il faut :

$$\begin{cases} t_{le} + t(\text{trait_e}) + d_{com} + t_{stock} + t(\text{trait_r}) + t_{lr} \leq R_{max} \\ t'_{le} + t'(\text{trait_e}) + d'_{com} + t'_{stock} + t'(\text{trait_r}) + t'_{lr} \leq R_{max} \end{cases}$$

Comme les temps de traitement et de communication sont non prévisibles, et surtout que leur fluctuation peut être importante dans le système distribué asynchrone considéré, il est impossible de raisonner à partir d'une valeur nominale de ces temps (excepté pour les temps de latence des cartes), et nous allons donc nous baser sur les

temps de traitement minimaux, qui représentent des bornes physiques incompressibles. C'est à partir de ces temps minimaux que la taille des tampons va être déterminée. Chaque objet stocké dans un tampon représente une durée de stockage égale à sa valeur nominale de présentation. Aussi, la taille des tampons vidéo et audio doit être respectivement T et T' avec :

$$\begin{cases} T = E\left(\frac{R_{\max} - t_{le} - t_{lr} - d_{\text{commu}} - t_{\min}(\text{trait_e}) - t_{\min}(\text{trait_r})}{t_{\text{nom}}}\right) + 1 \\ T' = E\left(\frac{R_{\max} - t'_{le} - t'_{lr} - d'_{\text{commu}} - t'_{\min}(\text{trait_e}) - t'_{\min}(\text{trait_r})}{t'_{\text{nom}}}\right) + 1 \end{cases}$$

où E représente la fonction "partie entière".

Bien sur, tout ceci suppose qu'un contrôle de débit est réalisé à l'émetteur. Ce contrôle de débit est d'ailleurs effectué par le processus émetteur lui-même qui joue le TSPN associé à l'application. De ce fait, il ne capture les données de l'application qu'au rythme précisé par le TSPN, qui est également le rythme de consommation de l'entité réceptrice de l'application.

Tout ceci a pour but de montrer que même si le stockage permet d'améliorer les problèmes des discontinuités, les tampons ont une taille limitée permettant de borner le retard de présentation de bout en bout. Pour rester en dessous de ce retard maximal, tout en stockant le plus longtemps possible pour éviter certaines discontinuités, il faudrait que les tampons vidéo et audio contiennent en permanence $T - 1$ et $T' - 1$ objets (ou T et T' occasionnellement). En effet, l'instant correspondant à l'arrivée des T et T' objets vidéo et audio peut également correspondre à l'instant pour lequel le retard de présentation est égal à R_{\max} (cela suppose que nous avons fait l'hypothèse que les T et T' premiers objets arrivent en un temps minimum). Ainsi, comme les vitesses des processus de production et de consommation sont équivalentes (aux problèmes de gigue et de dérive près), les niveaux de remplissage des tampons vidéo et audio doivent être respectivement $T - 1$ et $T' - 1$.

1.2.2. Mécanisme de résorption des retards

Toutefois, si l'application de présentation prend du retard, il faut pouvoir le résorber. L'application s'aperçoit qu'elle est en retard par rapport à l'émetteur lorsqu'elle ne peut plus stocker une donnée qui arrive. Dans ce cas là, le retard est résorbé en provoquant la perte de la donnée la plus ancienne du tampon (car sa sémantique temporelle par rapport au retard n'est plus valide). De cette façon, la partie présentation de l'application saute une donnée, ce qui lui permet de combler le retard qu'elle a pris.

Malheureusement, cette perte induit une rupture de la continuité du flux. Aussi, il a été proposé de résorber ces retards en faisant ralentir l'émetteur : au lieu d'émettre une donnée tous les t_{nom} , il n'émet une donnée que tous les t_{max} , ce qui permet à l'application réceptrice de petit à petit réduire son retard. Lorsqu'elle a enfin un retard acceptable, la cadence de l'émetteur peut de nouveau être augmentée. En asservissant ainsi la vitesse de l'émetteur sur la cadence du récepteur, on arrive à contrôler le retard de présentation. Toutefois, cette méthode s'est avérée expérimentalement très peu efficace (le temps de résorption d'un retard étant très grand), et elle a dû être abandonnée.

Limitations :

1. L'algorithme de contrôle du retard de présentation par perte ne garantit pas un respect fort de la borne R_{\max} qui peut être dépassée. En effet, les pertes sont provoquées lorsqu'une donnée arrivant ne peut être stockée dans les tampons. Cependant, avant même que cette condition soit vraie, il est possible que la borne R_{\max} soit dépassée. Par exemple, lorsque le tampon vidéo contient $T - 1$ images, et que l'on en stocke une

Tième, comme T a été calculé à l'aide de la fonction partie entière, on peut penser que le retard maximum n'est pas encore atteint. Toutefois, avant même que l'image suivante ($T + 1$ ième) n'arrive (si la partie présentation de l'application n'a pas consommé une image), la borne R_{max} sera atteinte et dépassée, et comme le temps n'est pas géré de façon explicite, il est impossible de s'en apercevoir avant l'arrivée d'une nouvelle image. En fait, ce mécanisme de contrôle des retards fait en sorte que le retard de présentation se situe toujours autour de R_{max} ; de plus, à cause de l'asynchronisme du système opératoire, du réseau, et des problèmes de désynchronisation des horloges sur un système distribué, il n'est pas possible de garantir le respect d'un retard maximum (même en gérant le temps de façon explicite).

2. Toutefois, lorsqu'une perte est provoquée par le mécanisme de contrôle du retard de présentation, il est certain que l'application a un retard ayant dépassé la borne R_{max} (pour les mêmes raisons que celles évoquées dans la remarque 1 et uniquement si l'hypothèse est vérifiée).

3. Cet algorithme conjecture que les T premiers objets seront reçus en un temps minimal, or cette hypothèse est très peu réaliste, ce qui amoindrit l'exactitude de cet algorithme en utilisation réelle. En effet, cette hypothèse permet d'éliminer les problèmes de gigue négatives au niveau des communications. Les stockeurs s'étant calés sur les premiers messages arrivés en un temps minimal, tous les autres paquets arriveront avec une gigue positive ou nulle. Cependant, l'hypothèse faite n'étant pas toujours vérifiée, des gigue négatives sont possibles. Dans ce cas là, cela se traduit par un remplissage prématuré des tampons ; ceci est analysé par l'application comme un retard de la partie présentation, et une perte est provoquée alors que la borne R_{max} n'avait peut être pas été atteinte ; ceci est dû au fait que la base temporelle implicite sur laquelle le mécanisme s'appuie n'est dans ce cas là pas correcte.

1.2.3. Contrôle de la synchronisation inter-flux dans le mécanisme de contrôle du retard de présentation

Jusqu'à présent, ce qui précède n'a traité que les problèmes de retard d'un flux unique par rapport à une base temporelle estimée du récepteur de l'application. Cependant, dans l'application de visioconférence PNSVS, la synchronisation inter-flux doit être assurée lorsque le mécanisme de contrôle des retards provoque une perte sur un des deux flux ; cette perte produite sur un flux doit obligatoirement induire la perte de l'objet associé sur l'autre flux, pour pouvoir garantir une synchronisation logique entre les données des flux audio et vidéo. Aussi, comme au niveau de l'émetteur l'image i correspond au son i , la perte de rang i provoquée par le récepteur sur un flux entraîne une perte provoquée sur l'autre flux pour la donnée de rang i également. Cependant, à cause des rendez-vous décalés, il n'est pas toujours possible, lorsqu'un retard non autorisé est observé, de perdre la donnée la plus ancienne du tampon. Par exemple, sur les figures II.2 et II.3, si un retard trop important est observé sur le flux vidéo, l'image 1 ne pourra être perdue, car elle est associée temporellement au son 1 qui a déjà été traité. Il faut donc tenir compte du décalage des rendez-vous, et sur ces exemples, l'image la plus ancienne contenue dans le tampon vidéo qui pourra être perdue est l'image 3. En revanche, sur le flux qui est traité en avance (ici le flux audio), ce problème ne se pose pas.

1.2.4. Optimisation de la gestion des retards

Le retard de présentation de bout en bout est un élément très difficile à maîtriser ; cependant, pour des raisons d'interactivité, il doit être le plus bas possible. Certaines études [JEF94a] ont montré que pour obtenir un bon niveau d'interactivité, il fallait que ce retard soit inférieur à 250 ms. Sur le matériel utilisé, il est impossible de rester en deçà de cette valeur, à cause en particulier des temps de latence des cartes multimédias. De plus, pour assurer une bonne qualité de présentation en réception, la visioconférence doit stocker les données pendant un certain temps. Or ce temps de stockage devra être d'autant plus court que les temps de traitements seront longs, et la qualité de présentation en sera affectée. Aussi, il est intéressant de pouvoir réduire les temps de

traitement à leur valeur minimale, et pour cela, il faut déterminer quels sont les traitements qui peuvent être raccourcis.

- Les temps de latence des cartes multimédias sont incompressibles, et les délais du système de communication ne peuvent être réduits par l'application. Ce n'est donc pas à ces niveaux qu'il faut intervenir.
- De même, les temps de traitement en réception ne peuvent guère être réduits. Les images sont des données atomiques traitées en une seule fois par la carte vidéo : il n'est donc pas possible d'intervenir sur le temps de décompression / présentation. Pour le son, l'élément atomique est l'échantillon, et chaque paquet contenant plusieurs échantillons peut donc être découpé. Ceci permettrait de réduire le temps d'écriture des données audio dans le tampon audio de la carte son, car plus la quantité de données est réduite, plus le temps d'écriture est court. Toutefois, le gain n'est pas apparent car il est compensé par les temps nécessaires aux appels système, et aux commutations induites par un plus grand nombre d'appels système.
- Par contre, il est possible d'intervenir sur les temps de traitement en émission. Pour la vidéo, pour les mêmes raisons qu'en réception, il n'est pas possible de réduire les temps de capture et compression. Par contre, le temps de traitement chez l'émetteur d'un paquet son dépend de la taille de ce paquet son : plus le paquet son est grand, plus le temps pour produire les données est grand (car les échantillons sont produits avec une cadence constante).

Par exemple, sur la figure II.3, les paquets sons de 100 ms (800 octets) nécessitent 100 ms pour être produits, et quelques 2 ou 3 ms de plus pour être empaquetés et émis. Ainsi, si le traitement sur une image (capture, compression, empaquetage, et émission) ne prend qu'environ 55 ms [DIA94a], le retard sera conduit par le flux audio. Sachant que les paquets audio sont non atomiques, il est possible de les découper, ce qui permet de gagner du temps sur le temps de production du paquet. Si on divise le paquet son de 100 ms en deux paquets son de 50 ms, le temps de traitement audio en émission passe à 52 ms, et celui sur la vidéo étant de 55 ms, cela permet de gagner près de 50 ms sur les traitements en émission. De plus, le retard est, dans ce cas là, conduit par le flux vidéo, et est de ce fait incompressible.

En ce qui concerne le contrôle du retard de présentation, pour conserver la synchronisation logique entre les données des deux flux (cf. 1.2.3.), provoquer la perte d'une image entraînera la perte de tous les paquets sons associés à cette image. De même, si une perte est provoquée sur le flux audio, cette perte concernera tous les paquets sons correspondant à une image. Pour ces raisons, le nombre de places sons sur une période du TSPN applicatif est un multiple entier du nombre de places vidéo sur cette même période. Le TSPN applicatif du récepteur résultant est représenté sur la figure II.4.

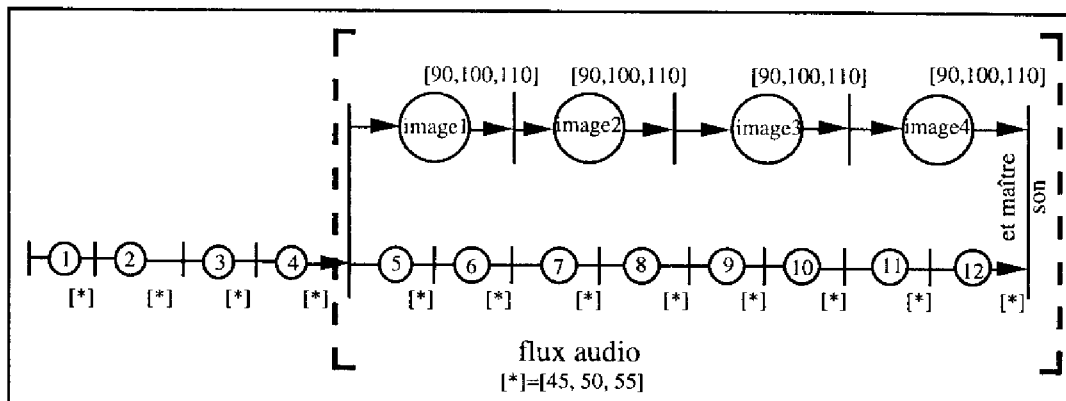


Figure II.4. TSPN applicatif tenant compte des améliorations pour le retard de présentation de bout en bout

1.3. Modélisation hiérarchique de l'application par des TSPN

[SEN94] et [DIA94e] ont montré que le modèle TSPN, que l'on peut rebaptiser TSPN de présentation, permet de décrire le comportement de scénarios de présentation multimédias. Nous venons de montrer qu'un seul modèle n'est pas suffisant pour décrire formellement l'ensemble de l'application. Le TSPN de présentation reste néanmoins intéressant pour l'utilisateur, car il permet de facilement exprimer les besoins en synchronisation d'un scénario multimédia. En revanche, à cause de la complexité du comportement du système opératoire et des cartes de présentation multimédias, le TSPN de présentation n'est pas utilisable directement en vue d'obtenir un réel ordonnancement temporel des processus. Aussi, nous avons montré qu'un autre modèle appelé TSPN applicatif est nécessaire pour décrire le comportement de l'application. Il est cependant essentiel que le TSPN applicatif puisse être déterminé à partir du TSPN de présentation accessible à l'utilisateur, et des autres paramètres de qualité de service.

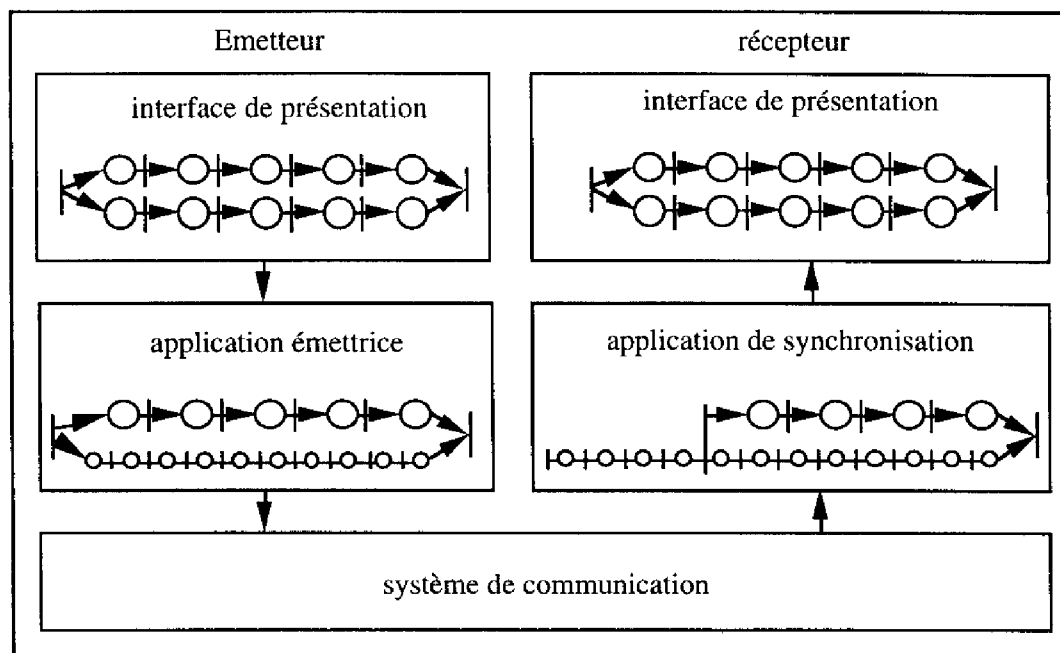


Figure II.5. Modélisation hiérarchique de PNSVS 1

Pour l'application de visioconférence PNSVS, qui est une application distribuée sur deux sites, [DIA93b] présente le modèle TSPN de présentation comme un modèle qui permet de spécifier les contraintes de synchronisation autant au niveau de l'émetteur que du récepteur. Le TSPN de présentation permet de caractériser la synchronisation à réaliser pour chaque site d'une application distribuée, le modèle étant le même pour chacune des machines distantes impliquées dans l'application. C'est le cas de PNSVS pour lequel le scénario des objets capturés et émis par l'émetteur est le même que celui des objets présentés au niveau du récepteur ; cela peut être, par exemple, le TSPN de présentation de la figure I.9 qui modélise pour PNSVS les contraintes de synchronisation. Toutefois, il a été montré que la synchronisation à mettre en oeuvre au niveau applicatif est différente de la synchronisation terminale. De plus, les fonctions de l'émetteur et du récepteur peuvent être très différentes du point de vue applicatif, et le modèle TSPN applicatif qui décrit l'application peut ne pas être le même pour l'émetteur et le récepteur. Par exemple, si le TSPN de présentation pour PNSVS 1 est celui de la figure I.9, le TSPN applicatif en réception est celui de la figure II.4. Néanmoins, les deux TSPN applicatifs (émetteur et récepteur) sont déterminés à partir du même TSPN de présentation. C'est ce que montre la figure II.5 qui représente les deux niveaux de la hiérarchie des TSPN modélisant l'application de visioconférence

PNSVS 1 lorsque les contraintes de synchronisation à respecter sont celles de la figure I.9.

2. Implémentation des mécanismes de synchronisation

Les mécanismes de synchronisation ont pour objectif de présenter les différents objets des flux audio et vidéo, en respectant leurs contraintes de synchronisation intra et inter-flux. En fait, le but est de pouvoir respecter les contraintes de qualité de service définies par un TSPN de présentation ayant la même forme que celui de la figure I.9. Ceci revient donc, étant donné les intervalles de présentation valides des différents objets, à écrire une application "*synchrone faible*" au dessus d'un système asynchrone.

De plus, tous les problèmes exposés ont été observés sur des machines Sun Solaris 2, et les solutions présentées utilisent la librairie des threads² normalisée dans le consortium de constructeurs Posix qui regroupe Sun, HP, Dec, etc. Ceci laisse penser que ces solutions seront portables sur les machines de plusieurs constructeurs, mais à ce jour, les solutions n'ont été testées que sur du matériel Sun Solaris 2.

2.1. Architecture de synchronisation

Avec les systèmes distribués asynchrones considérés, tous les composants sont asynchrones : supports de communication, systèmes opératoires et cartes de présentation multimédias. Aussi, les opérations de synchronisation doivent être réalisées au plus haut niveau de l'architecture du système de visioconférence, c'est-à-dire au niveau de l'application chez le récepteur. En effet, réaliser des opérations de synchronisation terminales au niveau des supports de communication ou des protocoles de communication est inutile, car les données synchronisées dans les couches basses seront redésynchronisées en passant dans le système opératoire et dans les cartes de présentation multimédias. De plus, l'application (partie utilisateur du système) est le seul endroit où le développeur peut intervenir sur le scheduling des processus ; dans le noyau, le développeur ne dispose d'aucun moyen de contrôle de l'ordonnancement des threads. La synchronisation sera donc faite dans la couche application. Toutefois, après être passées dans l'application de synchronisation, les données sont envoyées sur les cartes de présentation multimédias. Les données traversent de nouveau le noyau, et subissent donc son asynchronisme ainsi que celui des cartes multimédias (figure II.6).

Le fait que les données multimédias repassent dans le noyau et les périphériques de présentation après avoir été synchronisées au niveau de l'application représente un risque de désynchronisation : il est impossible de contrôler le comportement de ces données dans le noyau et sur les cartes, après leur synchronisation applicative terminale. De plus, les médias audio et vidéo sont traités par des cartes différentes et gérées par le noyau de façon indépendante, ce qui ne permet pas d'assurer un quelconque contrôle de la synchronisation pouvant exister entre deux flux. Toutefois, comme ces cartes sont gérées par le noyau, et à l'aide d'interruptions plus prioritaires que les applications temps partagé de l'utilisateur, et que les tâches réalisées par le système après la phase applicative de synchronisation ne sont pas très importantes, il est possible de conjecturer que les temps de latence des cartes de présentation multimédias sont constants, ou du moins que la gigue maximale qu'elles peuvent subir n'est pas suffisante pour perturber la synchronisation réalisée par l'application. Cette hypothèse est d'ailleurs vérifiée sur le matériel qui a été testé. Cependant, la solution idéale consisterait à disposer d'une carte unique de présentation multimédia intégrant de façon

²Un thread est un flot de contrôle avec un contexte réduit à l'intérieur d'un processus. Une bonne traduction du terme "thread" serait "processus léger". Toutefois, dans les dernières versions des systèmes UNIX (comme Solaris 2), un processus léger est une toute autre entité du système opératoire. Aussi pour ne pas prêter à confusion, le terme "thread" sera employé dans la suite de cette thèse.

synchronisée les traitements de l'audio et de la vidéo. Tant que de telles solutions matérielles n'existeront pas, le résultat visuel et auditif de la synchronisation applicative restera à la merci de la non véracité de notre hypothèse.

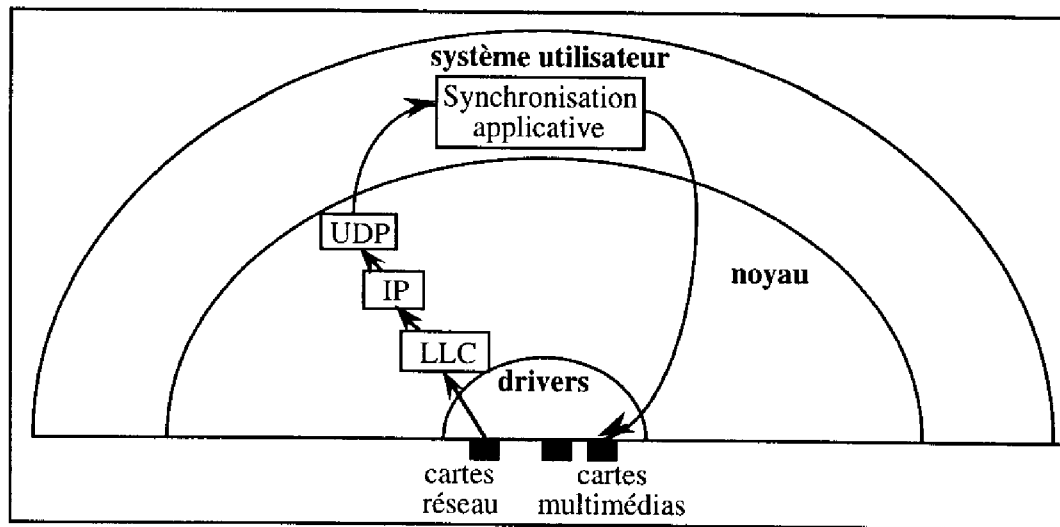


Figure II.6. Transit des données dans le système opératoire

Avec cette hypothèse sur les cartes de présentation multimédia, la tâche de synchronisation se réduit donc à des opérations réalisées au niveau de l'application. Cependant, cette application qui est écrite dans l'espace utilisateur du système opératoire doit être synchrone faible, et doit donc respecter des temps de présentation maximum. Or, le système opératoire étant asynchrone, aucun temps de traitement maximum n'est garanti en classe temps partagé. Pour pouvoir assurer des présentations qui ne dépasseront pas leur borne maximale, il est nécessaire d'utiliser des processus dont la priorité est supérieure à celle des tâches système, et de disposer d'un système interruptible. En Solaris 2, cette classe de scheduling s'appelle "Real Time" (RT). Toutefois, même en classe RT, les processus du système ne possèdent que peu des caractéristiques d'un processus temps réel. Leur attribut important est leur classe de priorité qui est maximale, et supérieure à la classe de priorité du système opératoire. Les différentes classes de priorités (ou classes de scheduling) du système Solaris 2 sont représentées sur la figure II.7.

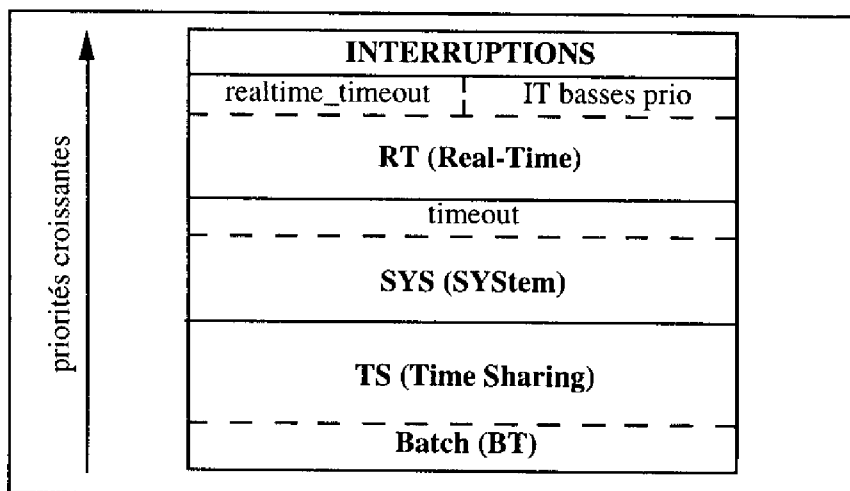


Figure II.7. Classes d'ordonnancement du système Solaris 2

Cependant, utiliser la classe de scheduling RT est très pénalisant pour les tâches système qui ne peuvent plus s'exécuter lorsque nécessaire et qui sont différées lorsqu'un processus RT s'exécute, et ceci risque de provoquer des dysfonctionnements du système [SUN94a][SUN94c]. Ainsi, une trop grande charge temps réel sur la machine qui perturberait le système, perturberait également l'application de visioconférence. Par exemple, si la machine est chargée par des tâches temps réel, les opérations de communication en classe système peuvent ne pas se faire. De même, une charge temps réel trop importante, utilisant beaucoup de `realtime_timeout`, peut perturber la gestion de certaines interruptions système de basses priorités, et l'hypothèse faite sur les temps de latence des cartes pourrait ne plus être vérifiée. Il faut donc réduire autant que possible la quantité des opérations faites en classe temps réel par l'application. De plus, si un processus temps réel fait un appel système, il perd son attribut temps réel, car l'appel système le fait passer en classe système. La classe de scheduling temps réel est incontournable, mais elle doit être manipulée avec prudence [OWE95a] [OWE95c].

L'architecture de l'application de visioconférence résultant de ces problèmes est représentée sur la figure II.8.

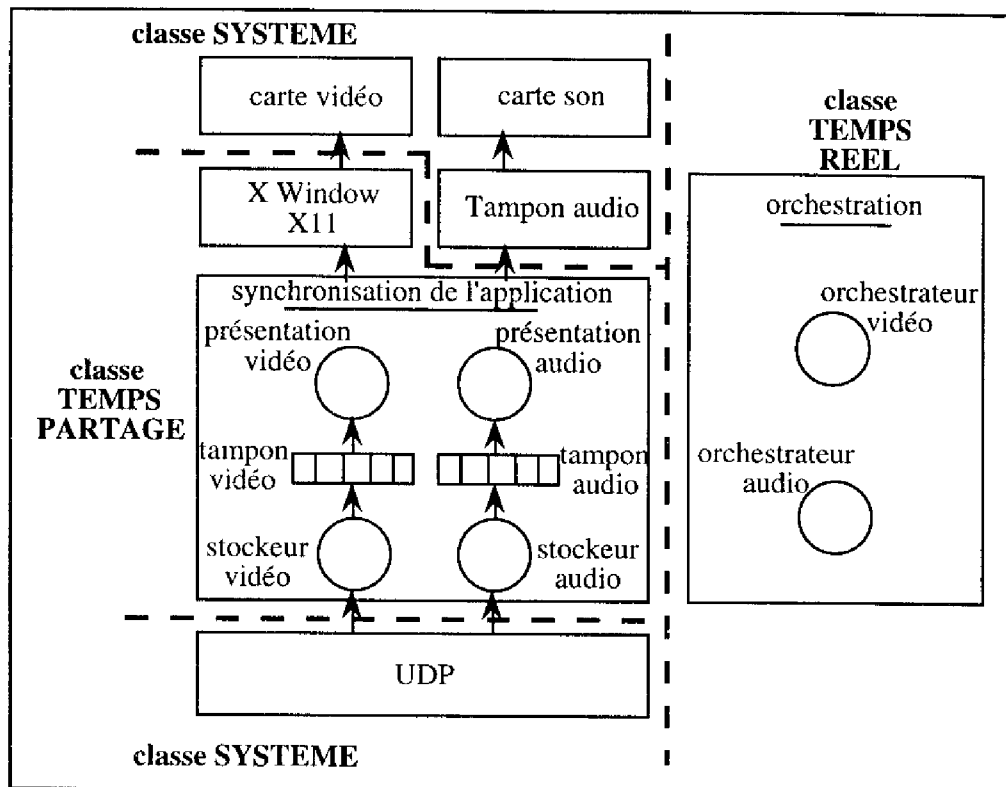


Figure II.8. Architecture de synchronisation pour la visioconférence

La figure II.8 représente les différents composants du système de visioconférence en associant à chacun d'eux leur classe de scheduling dans le système opératoire. De plus, cette figure montre le découpage de l'application en sous tâches, avec :

- les stockeurs audio et vidéo qui récupèrent les données transitant sur le réseau, et qui les stockent dans les tampons audio et vidéo. Ces tampons servent à stocker temporairement les données arrivant du réseau sans aucune garantie de synchronisation, et à les conserver suffisamment longtemps pour résoudre les problèmes de gigue ;

- les processus de présentation audio et vidéo temps partagé qui réalisent les opérations et les traitements nécessaires aux présentations des sons et des images vidéo ;
- les processus d'orchestration temps réel qui jouent le scénario de synchronisation.

2.2. Algorithmes de synchronisation

2.2.1. Algorithme de synchronisation intra-flux

L'algorithme de synchronisation intra-flux (audio ou vidéo) met en concurrence 3 processus, implémentés à l'aide de threads : le processus de stockage (audio ou vidéo), le processus de présentation (audio ou vidéo), et le processus d'orchestration (audio ou vidéo).

L'approche de synchronisation proposée dans cette partie apporte une solution au problème de la synchronisation en environnement asynchrone. Cette approche est diamétralement opposée à celle qui est utilisée dans les systèmes temps réel [JEF94a][COU94a][JEF92]. En effet, les systèmes temps réel garantissent des bornes supérieures aux temps d'exécution des opérations, et tous les algorithmes temps réel reposent sur cette propriété (même si le respect de cette borne n'est garanti que pour des processus de calcul pur, et donc ne peut être utilisé dès que le processus fait des entrées/sorties [BAK94]). Malheureusement, dans un système asynchrone cette borne supérieure n'existe pas. La seule borne qui existe est la borne inférieure car elle est physique (physiquement, à cause de la puissance d'une machine, une opération prendra un temps minimal incompressible). Tout l'algorithme de synchronisation présenté repose sur cette borne minimale des temps de traitement. En effet, le processus de présentation repose sur le fait qu'il peut s'exécuter en un temps minimal, et calcule ses valeurs d'attente sur les durées minimales ; comme les temps de traitement seront au moins égaux à ces valeurs minimales, le processus de présentation respectera sa borne minimale. Par contre, pour que le processus de présentation respecte le temps de présentation maximal, il a fallu lui associer un timer RT de façon à faire en sorte que la fonction d'interception du timeout soit la plus prioritaire du système, et assurer ainsi qu'elle ne subira pas les méfaits de l'asynchronisme système. De cette manière, les temps d'exécution des processus d'orchestration et d'interception du timeout sont connus et sont égaux au temps minimal physique de traitement de ces instructions, et ainsi, il est possible de ne pas dépasser le temps maximal de présentation des objets multimédias.

L'algorithme de synchronisation intra-flux met en concurrence des processus (implémentés à l'aide de threads), qui, pour se synchroniser, doivent accéder à des variables de synchronisation. La figure II.9 représente l'architecture des variables de synchronisation de PNSVS³. En fait, pour synchroniser deux threads, des variables conditions sont utilisées ; ces variables conditions consistent en une variable qui sera testée et/ou positionnée par les threads qui sont en attente d'un rendez-vous. Toutefois, pour éviter les conflits d'accès, des mutex (variables d'exclusion mutuelle) permettent d'accéder à cette variable en exclusion mutuelle. Enfin, pour permettre au premier thread arrivé au rendez-vous d'attendre le second, une variable condition intègre une file d'attente. Ce mécanisme est également utilisé entre les threads de stockage et de présentation ; dans ce cas, c'est une valeur d'état associée au tampon des données audio ou vidéo qui est utilisé comme variable.

³Les variables de synchronisation intra-flux, mais aussi inter-flux sont présentées, car la partie 2.2.2. présentera plus tard l'algorithme de synchronisation inter-flux, et se référera, pour cela, à la figure II.9.

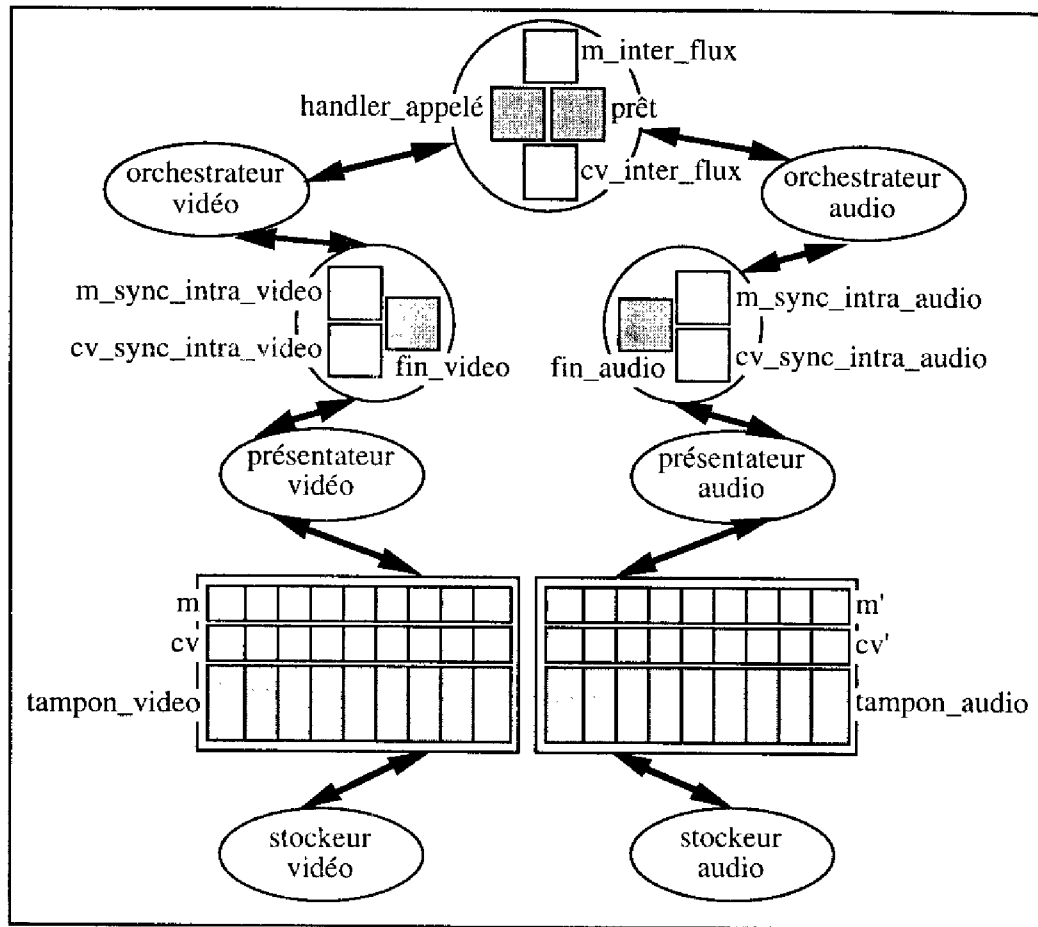


Figure II.9. Architecture des variables de synchronisation de PNSVS 1

Les variables et l'algorithme de synchronisation intra-flux qui sont maintenant présentés correspondent à la synchronisation intra-flux vidéo. Cependant, les mécanismes proposés sont les mêmes pour la partie audio ; seuls les noms des variables sont différents.

Les variables essentielles pour l'algorithme de synchronisation intra-flux vidéo sont les suivantes :

le numéro de l'image en cours de présentation ;
 un tableau m de mutex pour protéger l'accès au tampon vidéo ;
 Un tableau de variables condition d'attente des données vidéo (dans le tampon) ;

un tableau de variable statut définissant l'état des cases du tampon vidéo, à savoir : "donnée non arrivée" et "donnée présente" (le statut est initialisé à "donnée non arrivée") ;

le tampon vidéo ;

une variable $fin_vidéo$ qui indique si la présentation est achevée ;
 un mutex $m_sync_intra_vidéo$ pour protéger l'accès à $fin_vidéo$;
 et une variable condition $cv_sync_intra_vidéo$ pour les rendez-vous entre le processus d'orchestration vidéo et le processus de présentation vidéo.

Les algorithmes suivants utilisent également les constantes suivantes :

- t_{min} et t_{max} qui sont les temps de présentation minimum et maximum pour une image ;

- $t_{\min}(\text{traitement}_1)$ est le temps minimum de traitement lorsqu'une image est présentée, compression et affichage exclus (cf. [OWE95c]).
- τ est la période entre deux phases de scheduling ; en effet, lorsqu'un LWP⁴ (cf. remarque 3) passe de l'état "sleeping" à l'état "ready", son nouvel état ne sera pris en compte par le scheduler que lors de la phase de scheduling suivante. Sur Sun, la fréquence de scheduling pour le scheduler temps partagé est de 100 Hz, soit une période de 10 ms. Par contre, la fréquence (résolution) du scheduler RT peut être fixée par l'utilisateur : soit τ_{RT} cette valeur ;
- $t_{\min}(\text{traitement}_2)$ est le temps d'exécution minimum des opérations qui seront réalisées après l'expiration du timer RT (cf. [OWE95c]).

Algorithme du processus de présentation vidéo (classe temps partagé TS) :

```

fin_vidéo ← Faux ;

/* présenter l'image courante */
verrouiller le mutex de m associé à l'image courante ;
Si le statut de l'image courante est "image présente" alors
    /* cas normal de présentation d'une image */
    repositionner le statut de l'image à "image non arrivée" ; /* ré-initialisation pour le prochain
    passage */
    libérer le mutex de m associé à l'image courante ;
    décompresser et afficher l'image courante ;
    attendre  $t_{\min} - t_{\min}(\text{décompression/affichage}) - t_{\min}(\text{traitement}_1)$  ;
sinon
    /* exception : l'image précédente va remplacer l'image courante */
    libérer le mutex de m associé à l'image courante ;
    attendre  $t_{\min} - t_{\min}(\text{traitement}_1)$  ;
fin si ;

/* attendre l'image suivante */
verrouiller le mutex de m associé à l'image suivante de l'image courante ;
si le statut de cette future image est "image non arrivée" alors
    bloquer le processus de présentation vidéo sur la variable condition de cv associée à la
    prochaine image ; /* il sera libéré soit par le processus d'orchestration vidéo si le timer
    temps réel expire, soit par le processus de stockage vidéo s'il reçoit l'image attendue à
    temps */
fin si ;

libérer le mutex de m associé à l'image suivante de l'image courante ;

/* signaler la fin de la présentation à l'orchestrateur vidéo */
verrouiller le mutex m_sync_intra_vidéo ;
fin_vidéo ← Vrai ;
libérer le processus d'orchestration vidéo bloqué sur la variable condition cv_sync_intra_vidéo ;
libérer le mutex m_sync_intra_vidéo ;

terminer le processus de présentation vidéo ;

```

Algorithme du processus de stockage vidéo (classe temps partagé TS) :

```

Répéter à l'infini :
    réceptionner un message ;
    sauvegarder le numéro de séquence de ce message ;
    stocker le message dans le tampon vidéo en fonction de son numéro de séquence ;
    verrouiller le mutex de m associé au numéro de séquence du message reçu ;

```

⁴LWP : LightWeight Process (processus léger).

```

positionner le statut de ce message à "donnée présente" ;
signaler au processus de présentation l'arrivée de cette donnée. Ce signal est envoyé
sur la variable condition de cv associée au numéro de séquence du message reçu, et
peut éventuellement libérer le processus de présentation vidéo ;
libérer le mutex de m associé au numéro de séquence du message reçu ;
fin de boucle répéter

```

Algorithme du processus d'orchestration vidéo (classe temps réel RT) :

Répéter à l'infini :

```

Armer le timer temps réel pour qu'il expire au bout de  $(t_{\max} - t_{\min}(\text{traitement}_2) - \tau_{RT})$  ;
créer le thread de présentation vidéo (lié et détaché : cf. remarque 2 suivante) ;
verrouiller le mutex m_sync_intra_vidéo ;
si (fin_vidéo = Faux) alors
    bloquer le processus d'orchestration vidéo sur la variable condition cv_sync_intra_vidéo.
    Il sera libéré soit par un signal du processus de présentation vidéo lorsque ce dernier
    aura fini la présentation de l'image courante, soit par l'expiration du timer temps réel qui
    déroutera l'orchestrateur vers la fonction d'interception "handler" ;
fin si ;

si (fin_vidéo = Vrai) alors annuler le timer RT (cette valeur de fin_vidéo indique que
l'orchestrateur a été libéré par le processus de présentation : c'est le cas normal) ;
fin si ;
libérer le mutex m_sync_intra_vidéo ;

incrémenter le numéro de l'image courante ;
fin de boucle répéter

```

handler (timeout) :

```

détruire le processus de présentation retardataire ;
libérer les mutex qui peuvent avoir été pris, à savoir les mutex de m associés à l'image
courante et à sa suivante ;
fin de handler

```

Notes :

1. La primitive d'armement d'un timer permet d'armer une alarme qui devra se déclencher au bout du temps indiqué. Cette alarme, si elle se déclenche, générera un "realtime_timeout" (cf. figure II.7), qui se traduit par un déroutement immédiat du processus (léger) vers la fonction d'interception.

2. Tous les threads sont créés liés à un LWP (THR_BOUND) et ce pour 3 raisons :

- Tout d'abord, car il existe dans un système tel Solaris 2.x deux schedulings [POW91][STE92][EYK92][SUN94a] : un au niveau du système qui ordonnance les LWPs (assimilables à des threads du noyau), et un au niveau de la librairie (et par processus) qui ordonnance les threads utilisateur (figure II.10). L'idéal aurait été de ne manipuler que le scheduling de la librairie des threads car il est plus facile à contrôler et surtout plus performant [OIK94]. Cependant, le scheduling des LWPs ne peut être désactivé, et il perturbe, d'un point de vue utilisateur, le scheduling de la librairie des threads. Aussi, pour ne manipuler qu'un seul niveau d'ordonnancement (comme dans [ADE94]), on lie chaque thread à un LWP.

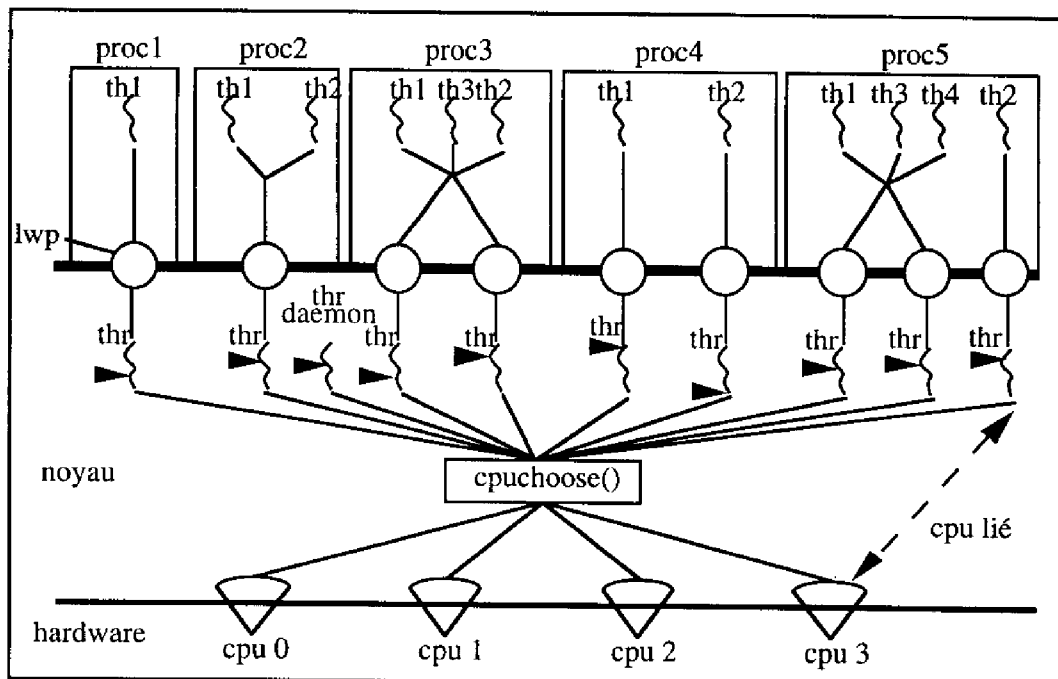


Figure II.10. Architecture de l'ordonnancement des threads dans Solaris 2

- De plus, en utilisant les threads liés, le niveau de concurrence est maximal et contrôlé par l'utilisateur. On évite ainsi les phénomènes de SIGWAITING [SUN94a][SUN94b] qui permettent l'ajustement automatique de la concurrence par le système, mais qui sont une source supplémentaire d'asynchronisme.
- Enfin, la classe de scheduling temps réel ne peut s'appliquer qu'à des LWPs, et n'existe qu'au niveau du scheduling système. Aussi, pour avoir des threads au comportement temps réel, il faut créer une association bijective entre un LWP temps réel et un thread [SUN94c].

Les threads sont également créés détachés (THR_DETACHED) pour éviter d'obtenir des threads zombie lorsqu'ils sont détruits et que leur code de retour n'est pas récupéré.

3. Dans l'algorithme de synchronisation intra-flux présenté, il y a décorrélation entre les comportements des processus de présentation et d'orchestration. Les contraintes temporelles sur le processus d'orchestration sont toujours garanties, car ce processus s'exécute en classe temps réel et ne fait appel qu'à des fonctions appartenant à des bibliothèques⁵ et ne fait donc pas d'appel système. Par contre, les contraintes temporelles du processus de présentation ne sont pas garanties à cause des multiples appels système qu'il réalise (et exécuter ce processus en classe temps réel serait inutile car les appels système font passer le processus en classe système, et lui font donc subir les problèmes d'asynchronisme). Cependant, le comportement asynchrone du processus de présentation ne pénalise pas l'application car le processus d'orchestration respecte parfaitement les contraintes de synchronisation lorsqu'il joue le TSPN de l'application. Ainsi, on obtient un comportement temps réel de l'application sur un système asynchrone, en décorrélant les processus au comportement temps réel des processus au

⁵L'appel à une fonction d'une bibliothèque ne modifie pas la classe d'ordonnancement d'un processus. En effet, le code même de ces fonctions ne contient pas d'appel à des primitives système, même si les actions réalisées dans ces fonctions peuvent ensuite provoquer des actions du système opératoire. Par exemple, la création d'un thread lié à un LWP engendre forcément la création d'un LWP et d'un thread noyau qui sont des ressources système. Toutefois, la fonction de la bibliothèque ne fera que positionner des variables de la structure de données du scheduler UNIX, et c'est lorsque le scheduler prendra la main que le LWP et le thread noyau seront créés. Ainsi, aucun changement de classe n'intervient lors de l'exécution d'une fonction d'une bibliothèque.

comportement asynchrone, i.e. les processus qui ne font que du traitement temps réel de ceux qui font des traitements asynchrones comme des entrées / sorties⁶.

Preuve de l'algorithme de synchronisation intra-flux vidéo :

- Pour le processus de présentation vidéo :

Ce processus réalise 3 opérations non atomiques :

- décompresser et afficher une image,
- le traitement₁,
- attendre $t_{\min} - t_{\min}(\text{décompression/affichage}) - t_{\min}(\text{traitement}_1)$.

Comme le système est asynchrone, on aura :

- $t(\text{décompression/affichage}) \geq t_{\min}(\text{décompression/affichage})$;
- $t(\text{traitement}_1) \geq t_{\min}(\text{traitement}_1)$;
- $t(\text{attente}(t_{\min} - t_{\min}(\text{décompression/affichage}) - t_{\min}(\text{traitement}_1))) \geq t_{\min} - t_{\min}(\text{décompression/affichage}) - t_{\min}(\text{traitement}_1)$.

Soit T le temps de traitement associé à une image. On a :

$$T = t(\text{décompression/affichage}) + t(\text{traitement}_1) + t(\text{attente}(t_{\min} - t_{\min}(\text{décompression/affichage}) - t_{\min}(\text{traitement}_1)))$$

donc :

$$T \geq (t_{\min}(\text{décompression/affichage})) + (t_{\min}(\text{traitement}_1)) + (t_{\min} - t_{\min}(\text{décompression/affichage}) - t_{\min}(\text{traitement}_1))$$

soit :

$$T \geq t_{\min}$$

Donc, le processus de présentation vidéo durera au moins t_{\min} , et la borne inférieure du temps de présentation d'une image sera donc respectée.

- Interception de l'expiration du timer :

La fonction d'interception du timeout vidéo doit arrêter les processus d'orchestration et de présentation vidéo associés à une image si le processus de présentation dure trop. Pour cela, il réalise les opérations :

- attendre l'arrivée du timeout temps réel qui doit arriver à $t_{\max} - t_{\min}(\text{traitement}_2) - \tau_{RT}$;
- prendre le processeur, ce qui induit une commutation ;
- traitement₂.

Or comme ce processus s'exécute en temps réel (et ne change pas de classe), il a une priorité supérieure à celle du système, et ne subira donc pas les problèmes de l'asynchronisme vus en 2. Aussi, il s'exécutera en un temps minimal, soit :

⁶Cette solution peut être également adoptée sur un système temps réel. En effet, les systèmes temps réel présentent des problèmes d'asynchronisme dès lors que des actions autres que du calcul sont effectuées (entrées / sorties, accès réseau, accès disque, etc.). Aussi, découper le problème en deux processus (un processus d'orchestration temps réel qui contrôle un processus de présentation au comportement asynchrone) semble être une solution intéressante.

$$t(\text{traitement}_2) = t_{\min}(\text{traitement}_2),$$

$$t(\text{timeout}) = t_{\max} - t_{\min}(\text{traitement}_2) - \tau_{RT},$$

Par contre, le temps avant la commutation de contexte $t_{\text{avant_commutation}}$ vérifie :

$$t_{\text{avant_commutation}} \leq \tau_{RT},$$

Soit T' le temps de traitement du processus de contrôle vidéo.

$$T' = t(\text{timeout}) + t(\text{traitement}_2) + t_{\text{avant_commutation}}$$

soit :

$$T' = t_{\max} - t_{\min}(\text{traitement}_2) - \tau_{RT} + t_{\min}(\text{traitement}_2) + t_{\text{avant_commutation}}$$

$$T' = t_{\max} - \tau_{RT} + t_{\text{avant_commutation}}$$

$$\text{or } t_{\text{avant_commutation}} - \tau_{RT} \leq 0$$

donc :

$$T' \leq t_{\max}$$

Donc, le processus d'orchestration pour une image durera au plus t_{\max} , et le temps maximum de présentation d'une image sera respecté.

2.2.2. Algorithme de synchronisation inter-flux

La synchronisation inter-flux est réalisée par les deux processus d'orchestration audio et vidéo. Le principe de cette synchronisation consiste à implémenter un rendez-vous entre les processus d'orchestration audio et vidéo, ce rendez-vous devant respecter les contraintes temporelles définies par la règle "et maître"⁷. Pour cela, les variables suivantes doivent être ajoutées aux précédentes (figure II.9) :

- une variable prêt qui donne l'état du rendez-vous entre les deux orchestrateurs ;
- un mutex $m_{\text{inter_flux}}$ pour protéger l'accès à la variable prêt ;
- une variable condition $cv_{\text{inter_flux}}$ pour le rendez-vous entre les deux orchestrateurs ;
- une variable handler_appelé qui indique si le timer temps réel a expiré.

Les algorithmes associés aux processus d'orchestration des flux audio et vidéo changent pour les places d'entrée de la transition inter-flux (les algorithmes des processus de présentation ne changent pas). Ainsi, pour les places son_{12} et image_4 du TSPN de la figure II.4, les algorithmes des processus d'orchestration et de présentation sont :

Orchestrateur vidéo :

```
Créer le thread de présentation vidéo (lié et détaché) ;
verrouiller le mutex  $m_{\text{sync\_intra\_vidéo}}$  ;
si ( $\text{fin\_vidéo} = \text{Faux}$ ) alors
    bloquer le processus d'orchestration vidéo sur la variable condition  $cv_{\text{sync\_intra\_vidéo}}$ .
    Il sera libéré soit par un signal du processus de présentation vidéo lorsque ce dernier
    aura fini la présentation de l'image courante, soit par l'orchestrateur audio, maître de la
    synchronisation inter-flux ;
fin si ;
libérer le mutex  $m_{\text{sync\_intra\_vidéo}}$  ;
```

⁷Ce principe reste valable dans le cas où la synchronisation inter-flux met en présence plus de 2 flux.

```

si (handler_appelé = Faux) alors
  /* le timer temps réel du processus d'orchestration audio n'a pas expiré : on n'est dans un
  cas normal de demande de rendez-vous avec l'orchestrateur audio */
  verrouiller le mutex m_inter_flux ;
  Si (prêt = Vrai) alors
    /* l'orchestrateur audio a déjà demandé le rendez-vous. Il ne reste plus qu'à lui
    signaler que le rendez-vous vient d'avoir lieu */
    libérer l'orchestrateur audio de la variable condition cv_inter_flux ;
  sinon
    /* l'orchestrateur vidéo est le premier à demander le rendez-vous ; il doit donc
    attendre l'orchestrateur audio */
    prêt ← 1 ;
    bloquer l'orchestrateur vidéo sur la variable condition cv_inter_flux en attendant
    l'orchestrateur audio ;
  fin si ;
  libérer le mutex m_inter_flux ;
fin si ;
incrémenter l'image courante ;

```

orchestrateur audio :

```

handler_appelé ← Faux ;
armer le timer temps réel pour qu'il expire au bout de  $(t_{\max} - t_{\min}(\text{traitement}_3) - \tau_{RT})$  ;
créer le thread de présentation audio (lié et détaché) ;
verrouiller le mutex m_sync_intra_audio ;
si (fin_audio = Faux) alors
  bloquer le processus d'orchestration audio sur la variable condition cv_sync_intra_audio.
  Il sera libéré soit par un signal du processus de présentation audio lorsque ce dernier
  aura fini la présentation de l'image courante, soit par l'expiration du timer temps réel qui
  déroutera le thread orchestrateur audio sur la fonction d'interception "handler" ;
fin si ;
libérer le mutex m_sync_intra_audio ;

si (handler_appelé = Faux) alors
  /* le timer temps réel n'a pas expiré : on est dans un cas normal de demande de rendez-
  vous avec l'orchestrateur vidéo */
  verrouiller le mutex m_inter_flux ;
  Si (prêt = Vrai) alors
    /* l'orchestrateur vidéo a déjà demandé le rendez-vous. Il ne reste plus qu'à lui
    signaler que le rendez-vous vient d'avoir lieu */
    libérer l'orchestrateur vidéo de la variable condition cv_inter_flux ;
    annuler le timer RT ;
  sinon
    /* l'orchestrateur audio est le premier à demander le rendez-vous ; il doit donc
    attendre l'orchestrateur vidéo (sauf si le timer expire) */
    prêt ← 1 ;
    bloquer l'orchestrateur audio sur la variable condition cv_inter_flux en attendant
    l'orchestrateur vidéo ;
    si (handler_appelé = Faux) alors
      /* le rendez-vous a eu lieu sans que le timer n'expire ; il faut donc l'annuler */
      annuler le timer RT ;
    fin si ;
  fin si ;
  libérer le mutex m_inter_flux ;
fin si ;
incrémenter le numéro de son courant ;

```

handler(timeout)

```

handler_appelé ← 1 ;
libérer d'éventuels threads bloqués sur cv_sync_intra_audio ;
libérer d'éventuels threads bloqués sur cv_sync_intra_vidéo ;
détruire les threads de présentation audio et vidéo ;

```

libérer d'éventuels threads bloqués sur `cv_inter_flux` ;
 libérer les mutex associés aux tampons audio et vidéo qui ont pu être verrouillés ;
 faire avancer l'orchestrateur vidéo à la première image de la période suivante si nécessaire ;

Remarque :

Le traitement₃ correspond à l'ensemble des opérations à réaliser après l'expiration du timer temps réel dans le cadre de la synchronisation inter-flux.

2.3. Définition complète des TSPN applicatifs

	paramètres fournis par l'utilisateur	paramètres mesurés par l'application	paramètres déterminés par l'application
TSPN applicatif émetteur	type de synchronisation * ‡	$t_{l_e}^\dagger$ $t'_{l_e}^\dagger$ $t_{\min}(\text{trait_e})^\dagger$ $t'_{\min}(\text{trait_e})^\dagger$ $d_{\text{com}_{\min}}^\dagger$ $d'_{\text{com}_{\min}}^\dagger$	Réseau de Petri du comportement applicatif de l'émetteur $[t_{\min}, t_{\text{nom}}, t_{\max}]_{\ddagger}^*$ $[t'_{\min}, t'_{\text{nom}}, t'_{\max}]_{\ddagger}^*$
TSPN applicatif récepteur	type de synchronisation * ‡	$t_{l_r}^\dagger$ $t'_{l_r}^\dagger$ τ_{RT}^\dagger $t_{\min}(\text{traitement1})^\dagger$ $t_{\min}(\text{traitement2})^\dagger$ $t_{\min}(\text{traitement3})^\dagger$ $t_{\min}(\text{décomp/affich})^\dagger$ $t_{\min}(\text{trait_e})^\dagger$ $t'_{\min}(\text{trait_e})^\dagger$ $t_{\min}(\text{trait_r})^\dagger$ $t'_{\min}(\text{trait_r})^\dagger$ $d_{\text{com}_{\min}}^\dagger$ $d'_{\text{com}_{\min}}^\dagger$	Réseau de Petri du comportement applicatif du récepteur $[t_{\min}, t_{\text{nom}}, t_{\max}]_{\ddagger}^*$ $[t'_{\min}, t'_{\text{nom}}, t'_{\max}]_{\ddagger}^*$

* paramètres hérités du TSPN présentation ‡ paramètres associés aux transitions
 † paramètres associés aux places
 ‡ paramètres associés aux arcs sortant des places

Figure II.11. Définition des TSPN applicatifs

Le chapitre 1 a donné une définition des caractéristiques du TSPN de présentation, qui est un réseau de Petri possédant des intervalles de validité temporelle sur les arcs sortant des places, et un type (associé à une sémantique de tir) sur les transitions inter-flux. Ce qui précède dans ce chapitre a montré que ce seul TSPN de présentation ne pouvait pas modéliser le comportement complet de l'application, et un TSPN applicatif pour l'émission et un pour la réception ont donc été définis. Toutefois, les algorithmes de synchronisation intra et inter-flux requièrent pour fonctionner un certain nombre de paramètres applicatifs comme par exemple $t_{\min}(\text{traitement}_1)$, $t_{\min}(\text{traitement}_2)$, et $t_{\min}(\text{traitement}_3)$, et il faut donc ajouter ces paramètres à la définition des TSPN de présentation pour obtenir la définition des TSPN applicatifs. Ces paramètres, qui permettent la dérivation du TSPN de présentation en un TSPN applicatif, sont

évidemment des paramètres qui sont associés aux processus : dans la modélisation des TSPN applicatifs, il sont donc logiquement associés aux places. Par exemple, si on considère le TSPN applicatif de la figure II.4, toutes les places audio et vidéo possèdent un paramètre $t_{\min}(\text{traitement}_1)$ et un paramètre $t_{\min}(\text{traitement}_2)$; Par contre, seule la place son_{12} - la place maîtresse de la transition inter-flux - possède le paramètre $t_{\min}(\text{traitement}_3)$ caractéristique de la synchronisation inter-flux. La figure II.11 indique tous les paramètres des TSPN applicatifs en précisant s'ils sont associés aux places, aux transitions ou aux arcs ; elle représente la définition complète des TSPN applicatifs.

3. Définition et négociation de la QoS dans PNSVS 1

3.1. Définition de la QoS

Les TSPN (de présentation et applicatifs) sont déterminés à partir des paramètres de qualité de service (QoS) fournis par les utilisateurs. Il existe deux types de paramètres de QoS : les paramètres dynamiques qui viennent d'être vus et qui influent sur les mécanismes de l'application, et les paramètres statiques qui n'influent pas sur son comportement dynamique. La figure II.12 présente tous les paramètres de QoS de l'application PNSVS 1, du point de vue de l'utilisateur.

	paramètres dynamiques de QoS	paramètres statiques de QoS
vidéo	nombre d'images/s gigue vidéo max nombre max de discontinuités	dimensions de l'image qualité de l'image
audio	gigue audio max nombre max de discontinuités	qualité du son
synchronisation	type de synchronisation dérive inter flux max	
retard de présentation	délai max	

Figure II.12. Définition de la QoS utilisateur

La QoS est négociée entre les deux interlocuteurs à l'établissement de la connexion. Cette négociation prend en compte, d'une part la qualité souhaitée par les utilisateurs (qui est fonction de leurs besoins mais qui peut aussi être fonction du coût de communication), et d'autre part de la puissance de traitement disponible sur la machine, la qualité du/des réseaux interconnectant les deux machines, et les formats d'échanges de données supportés par ces machines (généralement ces formats dépendent des cartes multimédias qui équipent les stations). Si les deux interlocuteurs sont d'accord sur une QoS que peuvent assurer les deux machines et le réseau de communication, alors la connexion est établie. Dans le cas contraire, l'application de visioconférence proposera aux utilisateurs une QoS plus faible qu'ils pourront accepter ou refuser.

Cependant, cette qualité ne peut pas être que statique. Elle doit être renégociable, c'est-à-dire qu'elle doit pouvoir être modifiée à tout moment, suite à un problème qui apparaît (surcharge due au lancement d'une nouvelle application par exemple) à

l'émetteur ou au récepteur, ou suite à une demande explicite de l'un des deux utilisateurs. La nouvelle qualité de service est négociée entre l'émetteur et le récepteur, de la même manière qu'au début de la visioconférence. Il faut noter que pendant toute la durée de la négociation, qui se fait sur un canal de signalisation annexe, la visioconférence continue de fonctionner avec la synchronisation en cours, c'est-à-dire, d'un point de vue formel, avec le dernier schéma TSPN sélectionné.

3.2. Renégociation de la QoS

Le problème consiste donc à définir une architecture et un protocole de renégociation ; en effet, la modification des contraintes de présentation implique des modifications dans toute la pile de l'application, sur chaque site, et la cohérence de l'ensemble oblige à être très attentif à la manière avec laquelle se déroulent ces changements. Il faut souligner que, lors de la renégociation de QoS, la partie sensible n'est pas la renégociation en elle-même (simple dialogue entre l'émetteur et le récepteur), mais la mise en application cohérente de la nouvelle QoS.

3.2.1. Contraintes

- Pour cela, nous avons mis en évidence un certain nombre de contraintes :
- par exemple, le choix du moment de commutation entre les différentes QoS est crucial car une donnée capturée avec une QoS donnée doit être restituée avec cette même QoS. En pratique, le choix de cet instant se ramène au choix d'un instant négocié de commutation. L'analyse montre qu'il doit être traduit en une valeur d'indice dans la numérotation des données. Une étude des mécanismes *Out-Of-Band* de choix de cet indice (sur le canal de signalisation annexe) nous montre que garantir l'arrivée de cet indice avant les données concernées est impossible (à moins d'arrêter le flux). Un codage de cette information devra donc être fait *In-Band*, dans les entêtes même des données ;
 - de plus, cette approche d'une commutation unique est incomplète : elle doit être unique du point de vue de l'utilisateur, mais est en fait multiple du point de vue de la machine. En effet, les différentes couches ne travaillent pas simultanément sur les mêmes données : si une couche traite la donnée d'indice n , la couche inférieure devrait être en train de préparer la donnée $n+p$ ($p>0$). Chaque couche doit donc gérer sa propre commutation.
De plus, à l'intérieur d'une même couche, du fait du traitement parallèle, un module de flux audio peut être en train de traiter une donnée d'indice n pendant que le module vidéo de la même couche finit de traiter la donnée précédente d'indice $n-1$. Chaque pile doit donc traiter sa propre commutation de QoS.
L'architecture de commutation de QoS et sa gestion doivent donc se calquer sur l'architecture multimédia de l'application (par couche et par pile) ;
 - quant à la commutation elle-même, elle s'avère très contrainte et doit être presque instantanée : en effet, elle se situe entre la présentation de 2 données à contraintes temporelles fortes. Or, cette nouvelle QoS nécessite sûrement des ressources différentes de la précédente (plus ou moins de tampons mémoire, une nouvelle fenêtre X, etc.), ressources dont la mise en place prend un temps en général non négligeable. Il faudra donc au maximum préparer ces ressources à l'avance afin de ne plus avoir qu'à les utiliser lors de la commutation effective.

3.2.2. Principe général du protocole

Le principe général de la renégociation de QoS dans PNSVS 1 peut être caractérisé par l'architecture de négociation représentée sur la figure II.13 :

1. Les deux négociateurs définissent le nouveau TSPN et mettent en place les ressources nécessaires à son application⁸.
2. Les négociateurs avertissent tous les modules de la future permutation (chez l'émetteur et chez le récepteur), leur demandant de se mettre en position de guet (de l'indice de commutation). Une fois les parties 1 et 2 terminées⁹, l'émetteur peut commuter dès qu'il veut.
3. Un module de capture de l'émetteur choisit les indices de commutation (en remarquant que des contraintes quant à ce choix peuvent exister : dans PNSVS 1, on ne peut commuter que sur la transition de synchronisation inter-flux), et les communique à chaque flux. Un codage de ces indices est désormais placé dans les en-têtes de tous les paquets¹⁰.
4. Chez le récepteur, le premier module utilisateur qui reçoit une donnée avec cette nouvelle information (généralement un de ceux de la couche utilisateur la plus basse), la décode et prévient tous les modules concernés en distribuant les indices de commutation.
5. Une fois prévenu (étape 4), chaque module compare son indice de travail courant et l'indice de commutation qui lui a été indiqué et commute au bon moment.

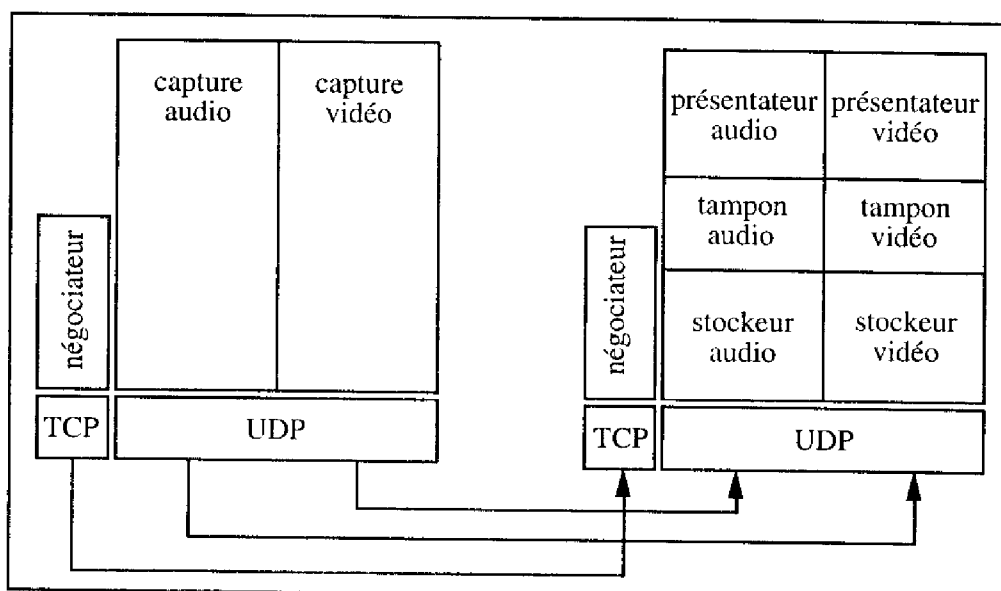


Figure II.13. Architecture de renégociation pour PNSVS 1

3.2.3. Détails du protocole de renégociation

Cette partie détaille maintenant le protocole de renégociation sur l'architecture présentée précédemment. Il faut noter toutefois que même si la demande de modification de QoS peut provenir de n'importe quel utilisateur, ce sera toujours, au niveau des entités protocolaires, le récepteur qui sera à l'origine de la demande de changement de QoS (dans la pratique, c'est souvent lui qui subit le fonctionnement le plus dégradé) ; toutefois, l'émetteur ne l'acceptera que s'il est capable de la satisfaire.

Comme expliqué dans la présentation générale de l'architecture, le protocole se divise en deux phases :

1. la négociation et mise en place des ressources,
2. la commutation module par module.

⁸Cette mise en place des ressources correspond à la contrainte de rapidité de la commutation.

⁹Un échange de message sera nécessaire entre les deux sites pour s'assurer que ces phases sont terminées.

¹⁰Il s'agit du codage In-Band imposé lors de l'étude des contraintes.

La négociation et mise en place des ressources

Cette partie du protocole est représentée sur la figure II.14. La notion de disponibilité des ressources a été introduite (cet aspect n'avait pour l'instant pas été présenté pour ne pas alourdir les explications, mais il doit rester présent à l'esprit).

Test et mise en place des ressources : souvent, le seul moyen de tester la disponibilité d'une ressource est de la demander et de regarder la réponse du système. Ceci ne gêne en rien l'algorithme présenté. Actuellement, cette part est assez réduite (le protocole transport UDP est sans réservation, la mémoire utilisée est assez faible et donc la réussite de la réservation presque automatique), mais peut prendre du temps.

Liens entre QoS1 et QoS2 : le récepteur s'assure que la QoS1 est compatible avec ses ressources avant de la demander.

L'émetteur regarde quelles sont ses ressources et renvoie QoS2, la plus proche possible de QoS1 qu'il puisse satisfaire : en général $QoS1 = QoS2$).

Pendant ce temps, le récepteur qui doit recevoir QoS2 met en place les ressources nécessaires à QoS1, quitte à en libérer si celles qu'il a réservées sont supérieures à celles dont il a besoin (dans ce cas, où la QoS2 nécessite moins de ressources que QoS1, on note $QoS2 < QoS1$).

Propriétés :

Lorsque le récepteur envoie START, toutes ses ressources sont prêtes.

Lorsque l'émetteur lit START, toutes ses ressources sont prêtes également.

Donc, lorsque l'émetteur lit START, toutes les ressources de l'émetteur et du récepteur sont en place : les phases 1 et 2 sont bien terminées.

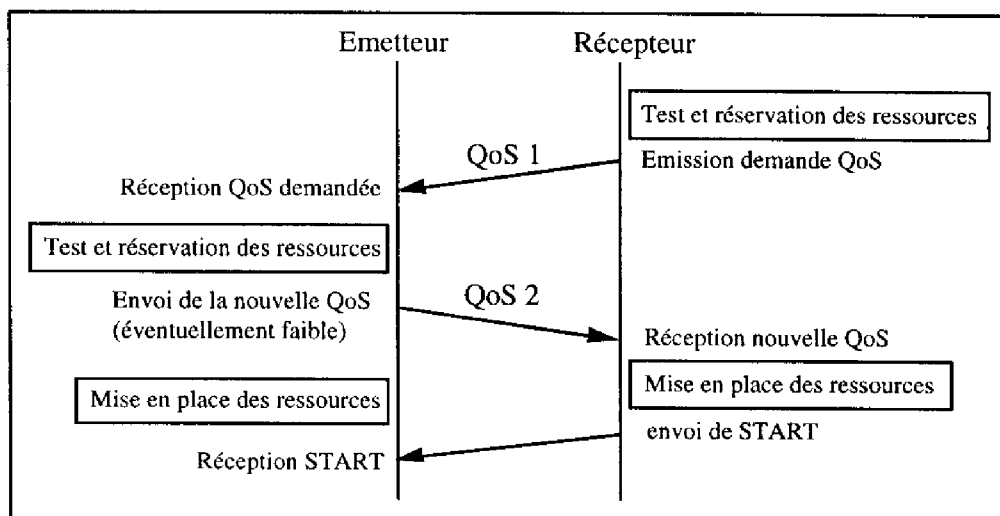


Figure II.14. Protocole de renégociation : négociation et mise en place

La commutation module par module

La partie commutation est très liée à l'architecture, puisque, comme précisé dans l'étude des contraintes, elle est gérée module par module.

Déroulement de la commutation :

1. Lecture de l'indice courant du flux maître.
Calcul de l'indice de commutation.
2. Codage et écriture de l'indice de commutation.

3. Insertion de l'indice de commutation et de l'indice courant dans l'en-tête de la donnée.
4. Décodage et propagation des indices de commutation.
5. Commutation de chaque module lorsque l'indice courant est égal à l'indice de commutation.

3.2.4. Le décalage d'indices

A cause des temps de latence différents suivant les cartes multimédias et les rendez-vous décalés, la présentation de données capturées simultanément implique qu'elles soient envoyées sur les cartes avec un certain décalage. Ce décalage se traduit par un décalage d'indices dans les données images et sons au niveau de la présentation (on envoie au même instant à la carte audio des données de validité temporelle t (et d'indice n) et à la carte vidéo (plus rapide) des données de validité temporelle $t - \delta$ (et d'indice $n - p$) car elles mettront moins de temps à arriver).

Lors du changement de QoS, on change la durée de présentation d'une donnée, mais comme le décalage temporel entre les cartes est constant, le décalage d'indices associés (noté p) est modifié (cf. formule (1) de la partie 1.1.2). Ainsi, si on prend comme exemple le cas d'une modification de QoS de 4 à 8 images par seconde, avec une différence des temps de latence des cartes audio et vidéo de 500 ms ($= t_1 - t_2$) en défaveur de la carte audio, alors les décalages qui devront apparaître entre les flux audio et vidéo sont :

- à 4 images par seconde : $N_d^4 = \text{arrondi}\left(\frac{500}{250}\right) = 2$

- à 8 images par seconde : $N_d^8 = \text{arrondi}\left(\frac{500}{125}\right) = 4$

Les TSPN correspondant à ces deux cas sont représentés sur la figure II.15.

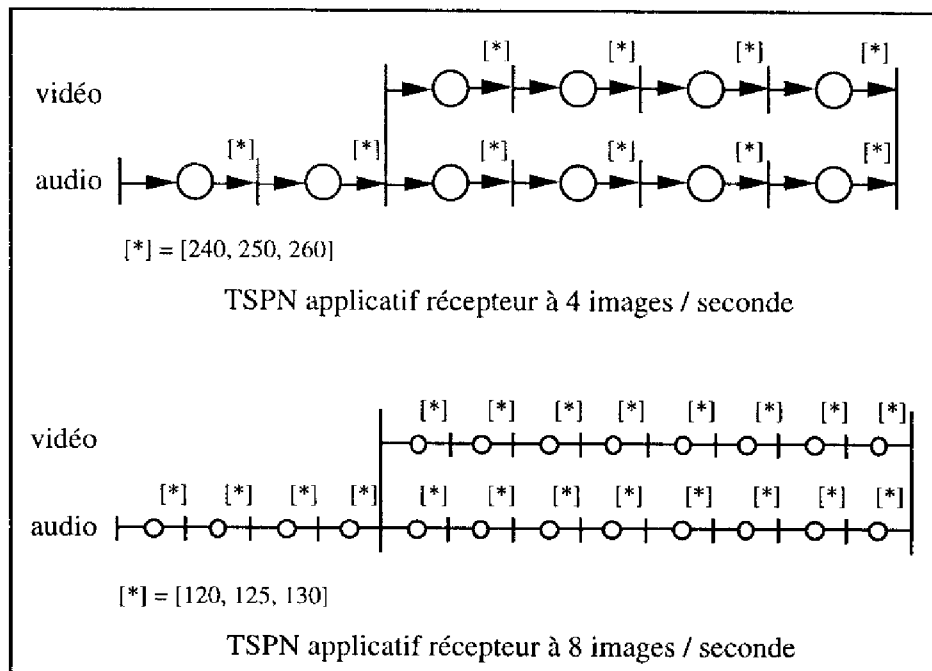


Figure II.15. Exemple de TSPN à 4 et 8 images par seconde

La figure II.16 donne le principe général de ce décalage. Elle illustre le passage de 4 images/s (avec un décalage audio/vidéo de 2) à 8 images/s (avec un décalage audio/vidéo associé de 4).

Sur ce schéma, les données ayant un même indice correspondent à des données ayant la même validité temporelle (en début d'application).

Comme on le voit dans la partie gauche du graphe de présentation du récepteur (figure II.16), le décalage entre flux est de 2 indices, et il amène donc à synchroniser l'image 0 avec le son 2. Pour changer de QoS (et donc de décalage), on le fera après l'image 3 et après le son 5.

Naturellement, on voudrait continuer la numérotation en séquence et donner à l'image suivante le numéro 4 et au son suivant le numéro 6. Mais avec 8 images/s, le décalage entre flux doit être de 4 et non de 2 (or l'écart entre le numéro de séquence du son - 6 - et le numéro de séquence de l'image - 4 - est de 2 unités, alors qu'après commutation de QoS, il devra être de 4).

Comme on le remarque sur cet exemple, on ne peut pas assurer à la fois la continuité dans les indices de chaque flux et le décalage entre flux.

On aurait pu envisager de commencer une nouvelle numérotation lors du changement de QoS, mais cela rendait difficile la gestion du transport. On a donc opté pour une numérotation avec discontinuité. Dans l'exemple de la figure II.16, on obtient un saut de 2 dans le flux audio. L'indice de détection de l'audio est de 6 ; cet indice n'apparaît pas sur la figure car il n'existe pas de donnée associée (à 7 non plus sur cet exemple), et c'est au moment où l'application souhaiterait présenter la donnée 6 qu'elle réalise qu'il faut commuter et faire le saut d'indice ; l'indice d'exécution est 8.

Pour plus de détails quant à ce protocole et à son implémentation en Solaris 2, le lecteur pourra se rapporter à [BOY96a].

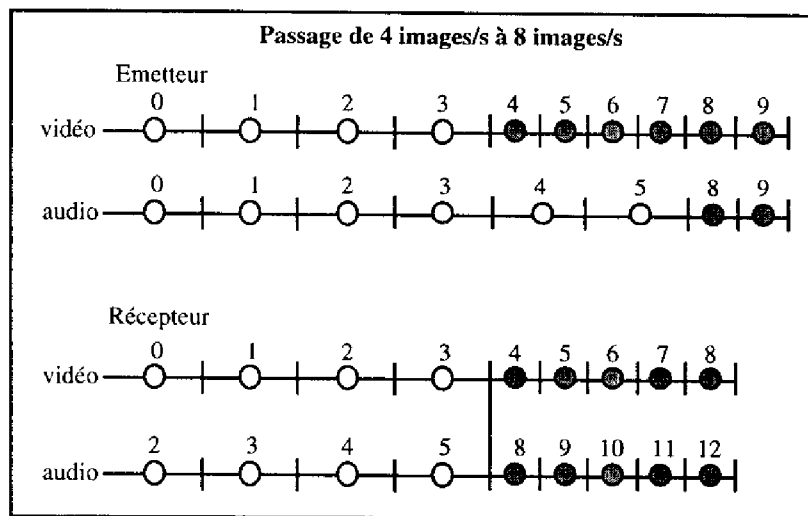


Figure II.16. Principe du décalage d'indice

3.2.5. Renégociation et contrôle du retard de bout en bout

En intégrant à l'application PNSVS 1 ces possibilités et cet algorithme de renégociation de QoS, l'algorithme de contrôle des retards, ou plus précisément la formule de calcul du remplissage moyen des tampons de réception doit être modifiée. En effet, la formule présentée au paragraphe 3 de ce chapitre considère que les objets audio et vidéo ont un temps nominal de présentation constant ; or l'objectif même du protocole de renégociation de QoS est de pouvoir modifier cette durée nominale.

Les solutions permettant de résoudre ce problème sont nombreuses. La solution qui a été retenue pour PNSVS 1 est d'associer à chaque tampon (audio et vidéo) un compteur qui totalise les durées de présentation nominale des objets stockés, et ce compteur, auquel on ajoute les temps de latence, de traitement et de communication, est

directement comparé au retard maximum autorisé R_{\max} , pour déterminer si des pertes doivent être provoquées. En ce qui concerne le dimensionnement des tampons lors d'une renégociation de QoS, si on passe d'une QoS pour laquelle les objets ont une durée nominale de présentation de t^1_{nom} à une QoS pour laquelle les objets ont une durée de présentation de t^2_{nom} , on remplacera dans les formules du paragraphe 3, t_{nom} par $\min(t^1_{\text{nom}}, t^2_{\text{nom}})$.

Si $t^1_{\text{nom}} < t^2_{\text{nom}}$, la réservation n'est pas optimale car tant que la visioconférence fonctionnera avec cette qualité de service, les tampons seront surdimensionnés. Cependant, cette dimension est requise au moment de la commutation.

4. Evaluation des mécanismes de synchronisation et de renégociation de la QoS de PNSVS 1

Les parties précédentes ont présenté des mécanismes de synchronisation et de renégociation dynamique de la QoS, et leur implémentation a été décrite. L'objectif de cette partie est maintenant d'évaluer ces mécanismes par l'intermédiaire d'une série de mesures. En particulier, les paramètres principaux sur lesquels vont porter ces mesures sont :

- la gigue ;
- la dérive inter-flux ;
- les pertes ;
- et le temps de commutation de QoS.

Ces mesures ont été réalisées avec le logiciel PNSVS 1 (au sein duquel ont été ajoutées des instructions de mesure) sur une plate-forme composée de 2 Sun Sparc station 10 bi-processeurs (une Sparc station 10-512 et une spar station 10-402).

4.1. Mesures de la gigue et de la dérive inter-flux

La première série de mesures a consisté à évaluer les mécanismes de synchronisation intra et inter-flux proposés dans la partie 2 de ce chapitre. Pour cela, il a été choisi d'utiliser le schéma de synchronisation modélisé par le TSPN de la figure I.9 (à savoir une visioconférence fonctionnant avec un débit de 10 images / s). Les résultats obtenus sont représentés sur les figures II.17, II.18 et II.19.

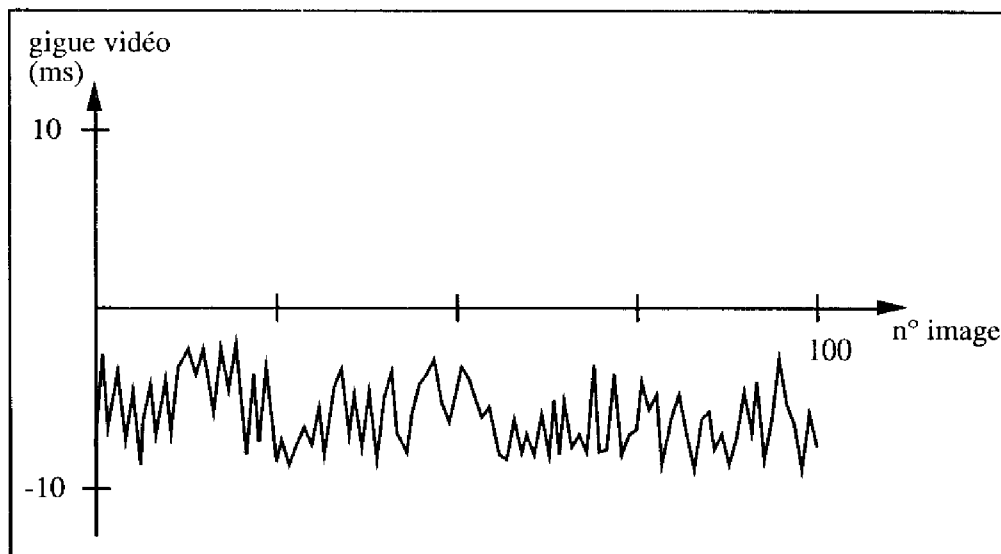


Figure II.17. Mesure de la gigue sur les images vidéo de PNSVS 1

La figure II.17 montre pour les 100 premières images (le reste de l'expérience étant conforme à ces 10 premières secondes), la gigue qui apparaît sur la présentation de chaque image (la partie audio ayant été désactivée). La gigue mesurée est la différence entre le temps de présentation réel et le temps de présentation nominal de chaque image. Cette courbe montre que la gigue maximale de 10 ms n'est jamais dépassée, mais est toujours négative ; en effet, dans ce cas, il n'y a pas de problème réseau (ni gigue ni perte) et donc les données sont disponibles au moment où le processus de présentation en a besoin. De ce fait, le mécanisme d'anticipation, qui rend possible l'arrêt de la présentation d'un objet dès que son temps minimal de présentation a été atteint, fonctionne systématiquement. Les variations qui apparaissent sont liées à l'ordonnanceur temps réel de Solaris 2.

La figure II.18 montre la même expérience appliquée au flux audio de PNSVS 1 (le flux vidéo ayant été désactivé). Comme pour la vidéo, les contraintes de présentation intra-flux sont toujours respectées, mais sont systématiquement positives. En effet, dans ce cas là, le tir de la transition intra-flux du TSPN est conditionnée par le signal de fin de consommation des données par le port audio. A l'échelle de temps que nous utilisons, cette consommation est toujours identique et égale à 100 ms. Les variations qui apparaissent sur la gigue sont liées au temps de prise en compte de cette information par le système.

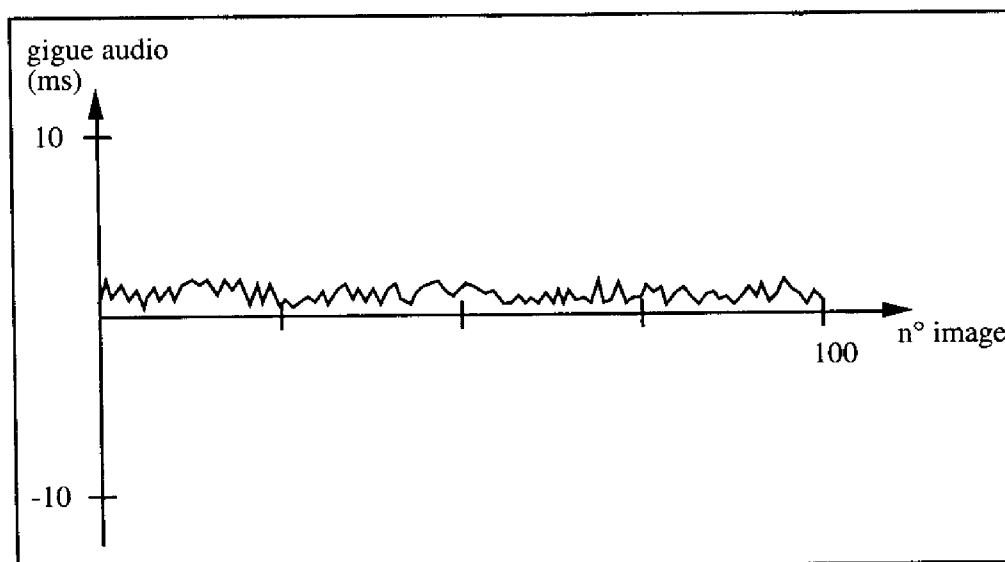


Figure II.18. Mesure de la gigue sur les objets audio de PNSVS 1

La figure II.19 montre la courbe représentant la dérive inter-flux en fonction du numéro d'image. Les Figures II.17 et II.18 ont montré que la gigue vidéo était négative et la gigue audio positive ; aussi, la dérive inter-flux (différence entre les dates de présentation d'un objet audio et de l'image associée) est positive. Cette dérive augmente d'ailleurs régulièrement sur chaque période (ceci est dû à l'effet cumulatif de la gigue), et elle est rattrapée lors de chaque synchronisation inter-flux (tous les 5 objets). Les contraintes de synchronisation inter-flux sont donc parfaitement respectées, puisque la dérive n'excède jamais 50 ms, la limite maximale étant 100 ms.

Il faut noter également que la charge machine n'a aucune influence sur ces résultats. En effet, la charge temps-partagé et système ne gêne pas les interventions des processus d'orchestration temps-réel de PNSVS 1, ce qui était prévisible. De plus, sur les machines bi-processeurs utilisées, il n'y a pas d'interférence entre les processus d'orchestration RT qui se répartissent sur les deux processeurs.

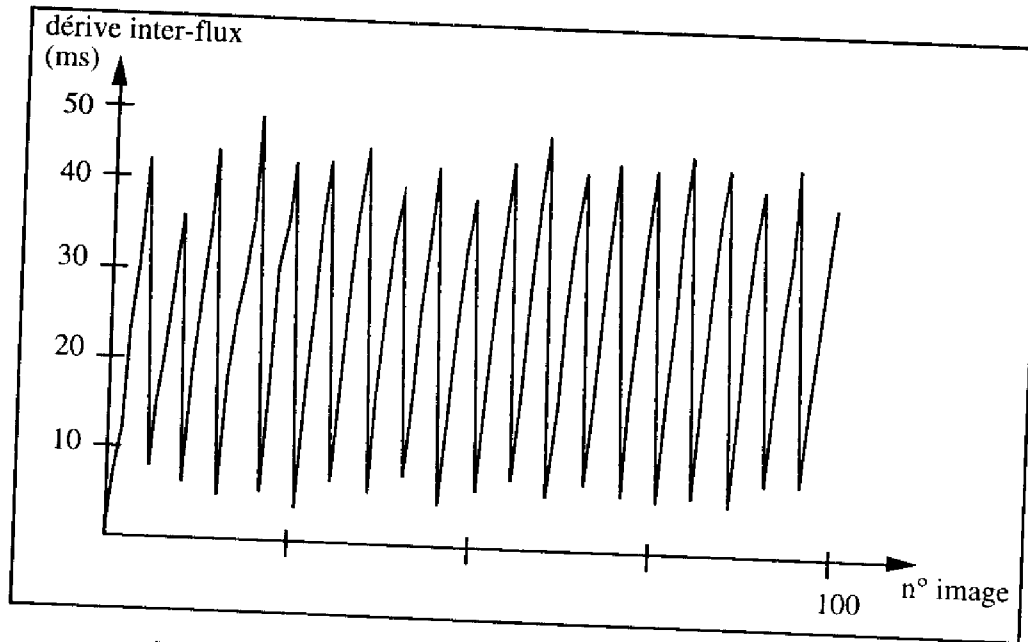


Figure II.19. Mesure de la dérive inter-flux dans PNSVS 1

4.2. Mesures de la gigue au dessus d'un réseau non fiable

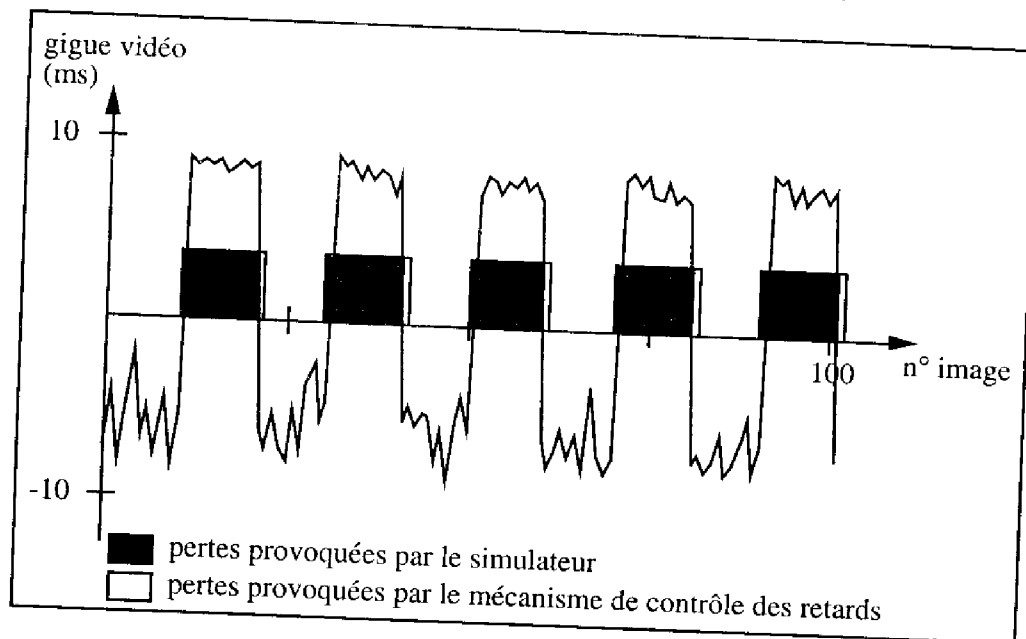


Figure II.20. Mesure de la gigue vidéo avec un réseau non fiable

Ce paragraphe va maintenant évaluer la gigue sur les données vidéo de PNSVS 1 lorsqu'un réseau non fiable est utilisé (les résultats sur l'audio suivent la même logique). Dans cette évaluation, le cas de la visioconférence modélisée par le TSPN de la figure I.9 a été utilisé, exceptée la période de synchronisation inter-flux qui a été augmentée à une série de 25 objets sur chaque flux. Un simulateur de réseau (développé dans ce but) a été utilisé. Ce simulateur génère 50% de pertes, en utilisant la répartition suivante : 10 objets sont transmis, puis 10 objets sont perdus, et ainsi de suite. Les résultats sont représentés sur la figure II.20. Les mesures de la gigue pour les paquets transmis

confirment les résultats représentés sur la figure II.18. Par contre, la gigue avoisine les 10 ms lorsque des pertes sont générées par le simulateur ; cette gigue correspond à la borne maximale de présentation des objets vidéo. Par contre, il faut noter qu'après une série de 10 pertes, le premier paquet transmis par le simulateur de réseau est déclaré perdu par le mécanisme de contrôle des retards ; en effet, l'accumulation de giges positives a fait augmenter le délai de présentation de bout en bout (c'est-à-dire le nombre d'objets stockés dans les tampons de réception de PNSVS 1).

4.3. Mesure des dates de présentation des objets de PNSVS 1 lors de changement de QoS

Dans cette expérience, nous avons voulu relever la date de présentation de chaque image de PNSVS 1 (les résultats sur l'audio étant similaires) dans un scénario programmé de modification de QoS. En fait, le scénario est le suivant :

- Tout d'abord, PNSVS 1 va fonctionner pendant 10 secondes avec un débit de 10 images / s ;
- Puis, pendant 10 secondes avec un débit de 20 images / s ;
- et enfin, pendant 10 secondes, avec un débit de 10 images / s.

Les résultats sont représentés sur la figure II.21, et sont tout à fait conformes à ceux escomptés ; en effet, la courbe obtenue est affine par morceaux (les problèmes de gigue, de l'ordre de quelques millisecondes étant invisibles sur un dessin à cette échelle). De plus, le passage d'un segment de droite à un autre (même avec une échelle plus petite) est très anguleux, et montre bien l'instantanéité de la commutation de QoS.

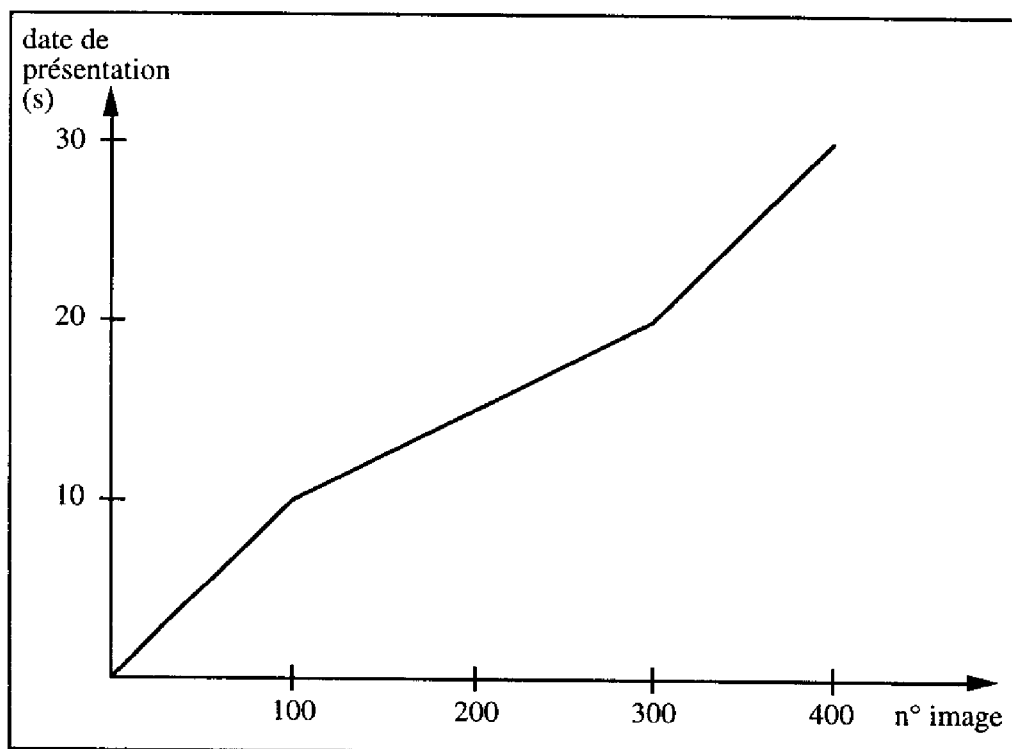


Figure II.21. Dates de présentation des images de PNSVS 1 lors de phases de commutation de QoS

Conclusion

Ce chapitre a présenté les mécanismes qui permettent de résoudre les problèmes posés par l'asynchronisme des systèmes distribués longue distance dans le cadre d'une application de visioconférence point à point. Ces mécanismes ont été décrits, ont été prouvés, et ont été modélisés à l'aide de réseaux de Petri. Pour ce qui est de la synchronisation multimédia intra et inter-flux, les spécificités du système normalisées par Posix 4 (obligatoires) ont été utilisées afin d'assurer un parfait contrôle des contraintes temporelles de présentation des unités de synchronisation des flux audio et vidéo. L'architecture proposée pour le module de synchronisation a d'ailleurs été développée avec comme souci de garantir les contraintes de présentation des objets multimédias, tout en économisant au maximum les ressources machines. A part quelques restrictions que l'on peut apporter sur le comportement de nos cartes d'acquisition et de présentation multimédia dont la prise en compte séparée par le système pourrait perturber la synchronisation finale, le résultat obtenu par notre application est tout à fait probant. L'analyse temporelle de l'application PNSVS 1 a été partiellement présentée au paragraphe 4, et les mesures qui ont été faites montrent que les contraintes temporelles de présentation des objets multimédias (autant en intra qu'en inter-flux) sont toujours respectées. [BOY96b] présente plus en détail toutes les mesures qui ont été réalisées sur PNSVS 1. En revanche, les mécanismes qui ont été développés pour assurer la continuité de présentation des flux et le contrôle du retard de présentation de bout en bout s'opposent. En effet, pour être efficace, le mécanisme de contrôle du retard de présentation doit parfois provoquer des pertes et donc des discontinuités sur les flux pour pouvoir récupérer un retard grandissant au dessus de la limite autorisée. De plus, le mécanisme de contrôle du retard de présentation ne peut pas garantir que le retard maximum acceptable ne sera pas dépassé : à cause des problèmes d'asynchronisme, ce mécanisme ne peut qu'assurer un service "best effort", i.e. il fait tout ce qu'il peut avec les ressources dont il dispose.

Toutefois, il a été observé que la réunion des trois familles de mécanismes permettant d'assurer une bonne synchronisation, peu de discontinuités et un retard de bout en bout acceptable engendre des baisses de qualité de service lorsqu'un réseau au taux de perte important est utilisé. Il en est de même lorsque le réseau possède un délai de transit qui fluctue beaucoup, et qui peut atteindre des valeurs importantes, car l'application considère comme perdues les données qui ne sont pas arrivées au bout d'un certain temps. Ainsi, lorsqu'une donnée est perdue, le mécanisme de synchronisation intra-flux qui l'attend va l'attendre jusqu'à la fin du t_{\max} de la donnée précédente. Or, si cette donnée a été perdue, cela conduit quoi qu'il arrive à une discontinuité dans la présentation ; non contente de cette dégradation, l'application a perdu du temps en attente ce qui a fait augmenter le retard de présentation de la donnée suivante. L'augmentation du retard peut ainsi conduire à de nouvelles pertes, soit par le mécanisme de contrôle des retards, soit par la gestion des synchronisations inter-flux "et-maître", et donc à d'autres discontinuités. Ainsi, la perte au niveau du réseau peut engendrer au niveau de l'interface de présentation de l'application plusieurs pertes. Il est clair que cette dégradation n'est pas acceptable car elle va rapidement conduire à des renégociations à la baisse de la QoS, et surtout une perte au niveau du système de communication ne devrait pas provoquer plus d'une discontinuité. Si on analyse, les effets d'une perte, on s'aperçoit que l'événement important qui enchaîne les problèmes est la perte de temps liée à l'attente d'une donnée qui n'arrivera jamais. Aussi, il semble indispensable de pouvoir déterminer au plus tôt les données perdues par le système de communication. C'est cette étude qui sera présentée dans le chapitre suivant, en utilisant un nouveau concept de protocole développé dans le groupe OLC du LAAS : les protocoles à ordres et fiabilités partiels.

Chapitre III

Utilisation d'un transport à ordre partiel pour optimiser la qualité de service d'une application de visioconférence

Introduction

Ce chapitre apporte une solution au problème de l'amélioration de la qualité de présentation dans les applications de visioconférence. Son intérêt sera montré dans le cas de PNSVS, en optimisant l'utilisation des ressources de communication ; ceci va permettre d'utiliser au mieux le temps, et donc la capacité de traitement mise à notre disposition par le matériel. Cette solution repose sur l'utilisation d'un service transport à ordre partiel, offrant de façon intégrée un service de fiabilité partielle permettant, tout en restant dans des normes de qualité de service acceptables pour l'utilisateur, de détecter les pertes au plus tôt, et de ne lancer les mécanismes de recouvrement que lorsque nécessaire. En fait, la solution proposée repose sur une architecture à 3 niveaux : transport à ordre partiel, pré-synchronisation et application de synchronisation terminale, dans laquelle les différentes opérations de synchronisation seront réalisées aux différents niveaux. Ce principe étend et généralise donc celui décrit dans le chapitre 2 pour lequel toutes les opérations sont réalisées au niveau le plus haut de l'application. Le plan de ce chapitre sera le suivant : tout d'abord, les problèmes rencontrés dans la première version de PNSVS seront présentés en détail. Puis, les principes d'un service de transport à ordre partiel seront sommairement exposés, et il sera montré comment il est possible d'utiliser ce service dans PNSVS 2 en gardant toujours à l'esprit la volonté d'optimiser la QoS. L'architecture préconisée sera ensuite détaillée, les services fournis par chacune des couches ou sous-couches (transport à ordre partiel et pré-synchronisation) seront décrits, et les algorithmes des mécanismes de ces deux niveaux seront proposés. Enfin, le modèle TSPN applicatif, vu dans le chapitre 2, sera étendu afin de pouvoir modéliser le comportement de chacune des nouvelles couches et sous-couches.

1. Les problèmes

Dans la conclusion du chapitre II, certains problèmes de la version 1 de PNSVS ont été exposés. Ces problèmes ont essentiellement deux origines :

- le premier problème provient des pertes qui sont occasionnées par le support de communication. Ces pertes peuvent vraisemblablement être acceptées si elles restent exceptionnelles. Toutefois, avec les principes utilisés dans la première version de PNSVS, ces pertes sont souvent la source de fortes dégradations de la qualité de présentation de l'application, et chaque perte engendre beaucoup plus qu'une simple discontinuité ; si une donnée est perdue par le réseau et si cette perte est tolérable par rapport à la qualité de service demandée par l'utilisateur, alors cette perte va conduire à une duplication de la donnée précédente, ce qui d'un point de vue présentation correspond à une discontinuité (ce qui semble être la dégradation normale et minimale dans ce cas précis). Cependant, avec les algorithmes de présentation décrits dans la première partie, comme les processus de présentation ne peuvent pas déterminer si cette donnée a été perdue ou si elle a seulement été retardée, ils attendent au maximum avant de lancer le traitement exceptionnel correspondant à une donnée perdue ou trop retardée (jusqu'au t_{\max} de la donnée précédente). Ainsi, l'application n'a pas pu recouvrir l'erreur engendrée par la perte, et elle a de plus perdu du temps à attendre la donnée à traiter (ce comportement est conforme à celui modélisé par le TSPN de l'application). A cause de cette perte de temps, ou à l'accumulation de ce type de pertes de temps, le retard de présentation de bout en bout augmente, ce qui peut conduire à l'intervention du mécanisme de contrôle du retard par perte, et peut donc provoquer de nouvelles pertes (le nombre d'objets perdus dépendant de la granularité de perte définie pour ce mécanisme). De même, le retard induit par l'attente d'une donnée provoque une augmentation de la dérive inter-flux, et lors du tir de la transition inter-flux, cela peut conduire à une accélération du flux en retard, et ainsi causer de nouvelles pertes sur ce flux. La perte d'une image qui devrait normalement ne causer qu'une discontinuité a donc des conséquences graves, puisque la prise de retard qu'elle engendre peut provoquer un nombre indéterminé de pertes causées par les mécanismes de contrôle du retard de bout en bout et de synchronisation inter-flux.
- le second problème est également un problème lié à des prises de retard, mais cette fois-ci au cours de la présentation des objets de l'un des flux audio ou vidéo. En effet, le paragraphe ci-dessus montre combien les retards peuvent être préjudiciables à la qualité de présentation de l'application. Aussi, il faut éviter au maximum que les temps de présentation des objets dépassent leur t_{nom} , ou du moins faire en sorte qu'en moyenne, les temps de présentation des données soient égaux à leur t_{nom} .

Pour palier à ces problèmes, il faut pouvoir détecter au plus tôt les pertes, et cette détection ne peut être faite qu'au niveau du support de communication. Ainsi, si les pertes sont détectées au plus tôt, cela permettra au processus de présentation de l'application de réaliser le traitement exceptionnel correspondant à une perte sans attendre, et ainsi de ne pas perdre de temps, et donc de ne pas risquer de provoquer d'autres pertes par des mécanismes annexes. De même, comme la partie non déterministe des temps de traitement des processus de présentation consiste à attendre la délivrance de l'objet suivant, si le support de communication assure une délivrance au plus tôt des objets reçus, cela permettra des gains temporels intéressants, car les processus de présentation ont toutes les chances de s'exécuter en un temps compris entre leur t_{\min} et leur t_{nom} (voir algorithmes de présentation du chapitre II), et aucun retard ne sera pris par l'application par rapport à son scénario de synchronisation idéal (celui qui respecte les durées nominales de présentation).

En fait, la délivrance des objets et la détection des pertes au plus tôt doivent permettre d'améliorer la qualité de présentation des flux multimédias, car elles permettent d'optimiser l'utilisation de la ressource temporelle pour un scénario de synchronisation décrit par un TSPN. Elles évitent donc que les mécanismes annexes pour le contrôle du retard et l'accélération de flux ne se mettent en marche. Ces deux mécanismes ne doivent être utilisés qu'en dernier ressort, par exemple lorsque la charge

machine est élevée, au point de ne plus permettre à l'application de s'exécuter ; l'utilisation de ces deux mécanismes qui nuit à la qualité de la présentation finale de l'application doit rester exceptionnelle.

2. Une solution basée sur un transport à ordre partiel

[AME94a] [CHA95a] [CHA95b] [DIA95] définissent un nouveau protocole de transport à ordre partiel comme un transport ayant pour but de délivrer à l'utilisateur les objets transitant sur une ou plusieurs connexions, en respectant un ordre donné. Cet ordre est n'importe quel ordre compris entre un ordre total (TCP) et le non ordre (UDP), et il peut s'exprimer, en particulier, sous la forme de compositions séries et/ou parallèles des objets à transmettre. Il s'avère alors qu'un tel ordre peut en particulier être celui décrit par l'automate du TSPN de l'application [DIA94c]. Ainsi, [AME94a] [CHA95a] [CHA95b] [DIA95] définissent cette délivrance suivant un ordre prédéfini comme une synchronisation logique définie par les informations multimédias.

De plus, cette nouvelle notion d'ordre partiel est complétée par la notion de fiabilité partielle. Par rapport aux problèmes rencontrés dans PNSVS 1, la notion de fiabilité partielle est essentielle. Cette notion est étroitement liée à la notion de qualité de service transport qui définit une qualité de service nominale, et une qualité de service minimale en dessous de laquelle le service demandé par l'utilisateur ne sera plus rendu. Par rapport à la notion de fiabilité, cette qualité de service minimale peut s'exprimer par un nombre maximal de pertes sur une séquence, et par un nombre maximal de pertes consécutives. Ainsi, en cas de perte acceptable, détectée lors de la réception d'un objet ordonné logiquement après l'objet attendu, l'objet initialement attendu est déclaré perdu au plus tôt (aucun essai de recouvrement n'est initié), et l'objet qui vient de parvenir à l'entité transport réceptrice est délivré à l'utilisateur (délivrance au plus tôt). Si par contre la perte n'est pas acceptable par rapport à la fiabilité définie par l'utilisateur, un certain nombre de retransmissions peuvent être essayées, ce nombre de retransmissions pouvant être paramétré. Tout objet reçu par l'entité transport réceptrice est donc délivré au plus tôt en accord avec l'ordre et la fiabilité partielle ; si cet objet n'est pas délivrable au regard de l'ordre partiel, il est vraisemblable qu'un problème a perturbé les objets qui le précèdent logiquement, et donc, si au regard de la fiabilité partielle les objets manquant peuvent être perdus, alors ils seront considérés comme perdus, et l'objet reçu ne sera pas retardé (délivrance au plus tôt)[AME93a] [AME93b].

En fait, il existe deux approches pour la gestion des principes de fiabilité partielle : une gestion média par média et une gestion par groupe de médias. Dans la gestion média par média, l'entité réceptrice d'un flux ne peut utiliser les mécanismes de fiabilité partielle que sur le flux qu'elle gère, et pas sur les autres flux de la multi-connexion. En revanche, avec une gestion par groupe de médias, l'entité réceptrice d'un flux peut, pour mettre en œuvre le principe de délivrance au plus tôt, provoquer des pertes sur d'autres flux au moment des synchronisations logiques inter-flux [DIA95].

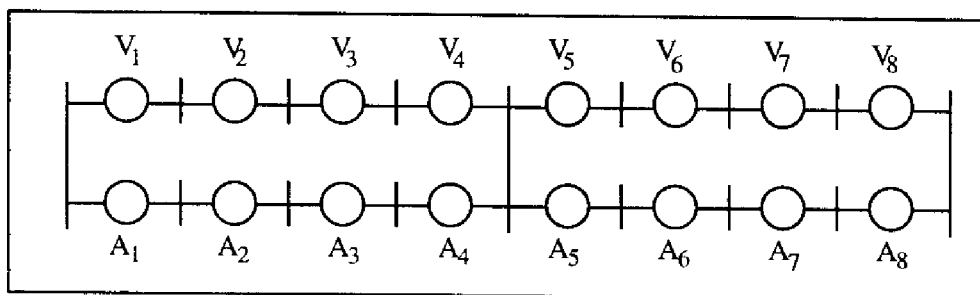


Figure III.1. Exemple d'ordre partiel

Prenons pour illustrer ces propos l'exemple de la figure III.1 qui représente le réseau de Petri des compositions série/parallèles pour une multi-connexion audio / vidéo.

Supposons que le nombre de pertes maximal acceptables sur un flux et par période est d'un élément. Supposons de plus que les objets V1, A1, A2 et A3 ont été reçus par l'entité transport réceptrice et délivrés à l'application. Supposons maintenant que l'entité réceptrice du transport reçoive l'objet V4 ; en respectant l'ordre partiel sur la connexion vidéo, la délivrance de l'objet V4 à l'utilisateur ne peut se faire, car il faudrait pour cela déclarer perdus les objets V2 et V3. Or deux pertes par période et par flux sont interdites. L'objet V4 est donc mémorisé en attendant. Si l'objet V3 arrive, il peut être délivré à condition de déclarer perdu l'objet V2. Pour ne pas ralentir les transmissions, le transport à ordre partiel délivre dès que possible les objets reçus et / ou mémorisés ; il déclare donc au plus tôt la perte de l'objet V2 (dès qu'il reçoit V3), et délivre en suivant l'objet V3, et l'objet V4 qui avait été stocké : principe de délivrance au plus tôt permis grâce à une perte au plus tôt. Ceci illustre en fait une gestion par média des principes d'ordre et de fiabilité partielle.

Nous allons maintenant illustrer ces deux mêmes principes par rapport à une gestion par groupe de médias. Supposons pour cela que l'objet V5 soit reçu par l'entité réceptrice du transport. Cet objet n'est pas délivrable au regard de l'ordre car A4 n'est pas arrivé, et qu'il y a une synchronisation logique qui apparaît entre les deux flux audio et vidéo après les objets A4 et V4. Aussi, si on autorise une gestion de la fiabilité par groupe de médias, l'arrivée d'un objet du flux vidéo peut provoquer la déclaration de pertes sur le flux audio, dans le cas où cela permet de délivrer plus rapidement certains objets reçus. Dans cet exemple et en mettant en oeuvre ce principe, l'arrivée de V5 provoque la déclaration de perte de A4 (ce qui est autorisé au regard de la fiabilité sur le flux audio), et l'objet V5 est délivré au plus tôt. Cette gestion de la fiabilité par groupes de médias nécessite donc que l'architecture de l'entité réceptrice du transport (figure III.2) intègre un gestionnaire de multi-connexions multimédias [CHA95d].

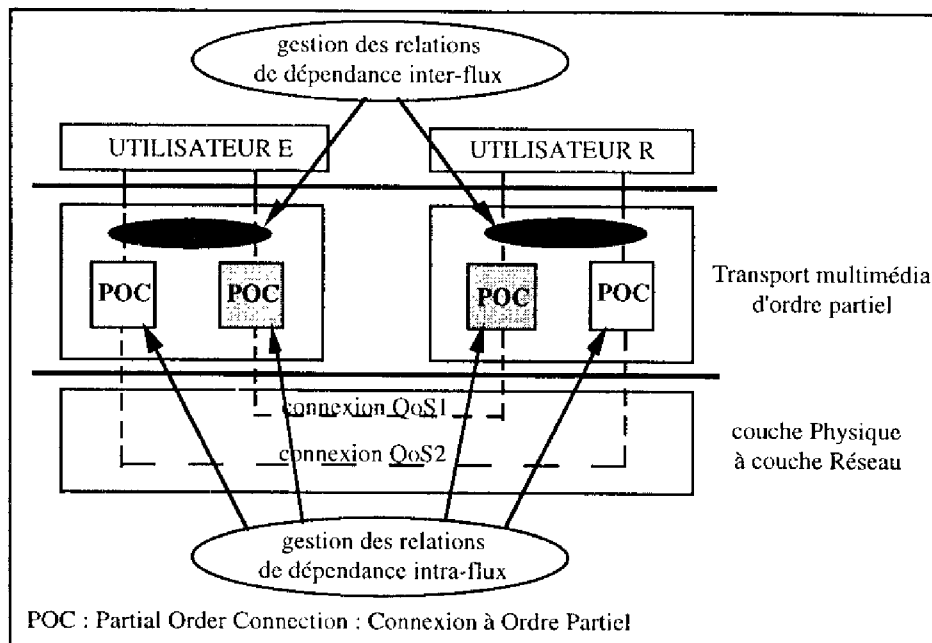


Figure III.2. Architecture des entités émettrice et réceptrice d'un transport à ordre partiel

L'application de visioconférence PNSVS respectera forcément le TSPN qui décrit les ordres et les temps de présentation des objets multimédias, et même si les objets ne lui étaient pas délivrés dans l'ordre, elle serait capable de les présenter dans le bon ordre (les objets étant toujours stockés dans le bon ordre par les processus de stockage). D'ailleurs, le transport à ordre partiel, s'il est intégré dans un système multi-couches asynchrone, ne peut pas garantir à l'utilisateur un ordre de délivrance à cause de

l'asynchronisme du système opératoire. En effet, même si le transport délivre un objet audio et un objet vidéo à l'utilisateur dans un ordre donné, et ce sur deux points d'accès (soit en UNIX deux descripteurs de fichiers) différents - comme décrit dans l'architecture de gestion d'une connexion multimédia à ordre partiel [CHA95d] - les indications de délivrance qui proviendront au processus de l'application utilisatrice peuvent arriver dans un ordre différent. C'est donc à l'application de gérer l'ordonnancement logique des données, mais cela n'apporte pas une charge de travail supplémentaire car l'ordonnancement logique est inclus dans l'ordonnancement temporel que doit garantir l'application. Toutefois, la notion d'ordre dans le service à ordre partiel est primordiale, car elle permet de détecter les ruptures de séquences dans les flux reçus, et tous les mécanismes d'optimisation liés à la fiabilité partielle en dépendent (autant en considérant un ordre partiel sur une connexion qu'un ordre partiel sur plusieurs connexions, car dans des applications comme PNSVS 2, comme dans de nombreuses applications, les flux qui transitent sur des connexions différentes ne sont pas indépendants).

En fait, il s'avère que bien que développé indépendamment, le principe de la fiabilité partielle (délivrances et pertes au plus tôt) définie dans le service transport à ordre partiel correspond tout à fait aux besoins de PNSVS 2. L'interconnexion n'est cependant pas immédiate, et les principes de l'ordre partiel, ses services et fonctionnalités doivent être affinés pour s'adapter parfaitement aux flux continus audio et vidéo qui seront utilisés par PNSVS 2. La partie suivante propose d'ailleurs une architecture pour la visioconférence PNSVS 2 construite sur un transport à ordre partiel. Toutefois, l'étude réalisée ne considère que des flux isochrones (audio et vidéo), et aucune considération quant au transport d'autres types de médias n'est évoquée.

3. Architecture

L'architecture requise pour exécuter PNSVS au dessus d'un transport à ordre partiel est décrite sur la figure III.3. Cette architecture ne positionne pas directement la tâche de synchronisation au dessus de l'ordre partiel dont le service rendu, notamment par rapport aux contraintes temporelles (à cause de l'asynchronisme du système), n'est pas complètement précis. Aussi, une sous-couche de pré-synchronisation temporelle a été ajoutée à l'application pour compléter les propriétés temporelles du service transport [OWE95c] [OWE96c] [OWE96e].

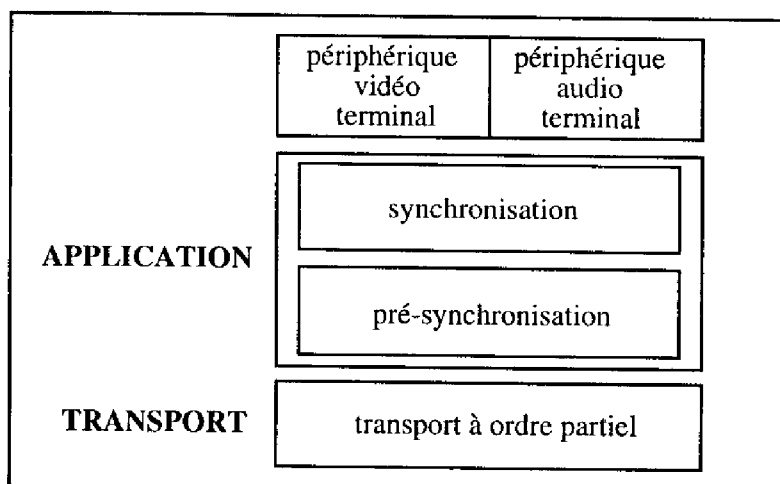


Figure III.3. Architecture de PNSVS 2

Cette architecture comporte principalement trois niveaux de fonctionnalités : le transport à ordre partiel, la sous-couche de pré-synchronisation, et la sous-couche de présentation synchronisée de l'application. La suite décrit les fonctions de chaque couche ou sous-couche : elle décrit pour chacune d'elles leurs actions sur la synchronisation (qui doit être assurée sur les objets de présentation de PNSVS 2), et justifie ce rôle par rapport à la position de la couche ou sous-couche dans le système opératoire asynchrone.

Les fonctions du transport à ordre partiel ont déjà été énumérées dans ce qui précède. En résumé, on peut dire que ce transport doit permettre d'ouvrir des connexions à hauts débits pour la communication des objets multimédias de l'application PNSVS 2. Ces connexions ont de plus une qualité de service propre en terme de débit et fiabilité et doivent, grâce aux principes de fiabilité partielle et de délivrance au plus tôt (basés sur un ordre partiel précisé par l'application), être suffisamment rapides pour ne pas ralentir l'application. Toutefois, le transport à ordre partiel ne garantit aucune contrainte temporelle sur les flux pourtant isochrones qu'il transporte ; le transport ne gère pas le temps de façon explicite, mais de façon implicite au travers de ses tampons en associant à chaque objet stocké une durée égale à sa durée nominale de présentation. Grâce à ce principe, le transport à ordre partiel peut déterminer s'il doit demander des retransmissions, et le nombre de retransmissions qu'il peut demander de sorte que le retard de présentation maximal autorisé par l'application ne soit pas dépassé ; le délai de transport maximal s'exprime d'ailleurs en terme de remplissage maximal des tampons. Cette gestion temporelle implicite par les tampons n'est pas très précise. Toutefois, une gestion explicite du temps n'apporterait pas une garantie en terme de précision, à cause de l'asynchronisme système, et serait donc plus pénalisante en charge machine qu'utile ; cette gestion explicite du temps ne serait qu'approximative.

A cause de cette gestion du temps en fonction du niveau de remplissage des tampons, le transport à ordre partiel seul ne peut détecter que très tard les longues séquences de pertes, et n'apporte dans ce cas là aucune optimisation par rapport à un transport comme UDP. Ces pertes en séquence ne peuvent pas être détectées sans une gestion explicite du temps (à l'aide d'une horloge). Aussi, la sous-couche applicative de pré-synchronisation a été ajoutée entre le transport et la synchronisation pour réaliser les optimisations temporelles que le transport ne peut effectuer, et ainsi améliorer la qualité de présentation de l'application. La sous-couche de pré-synchronisation gère donc le temps de façon explicite, et apporte un contrôle temporel sur les données délivrées ou perdues par le transport ; elle permet de détecter les longues séquences de pertes, les problèmes de dérive du réseau et les problèmes d'asynchronisme du système de communication. Elle ne permet pas des gains de temps aussi importants que les mécanismes de délivrance et perte au plus tôt du service transport, mais évite d'avoir à provoquer des pertes de temps lorsque des phénomènes perturbateurs viennent gêner le fonctionnement normal du transport. Elle permet même une légère optimisation temporelle, car ayant connaissance des temps de transferts des données entre la sous-couche pré-synchronisation et la sous-couche de synchronisation, elle peut les déduire avant de déclarer une donnée perdue : cette optimisation est basée sur le même principe que la perte au plus tôt des objets par le transport, sauf que la décision est prise au regard d'une horloge, plutôt que par rapport à un ordre logique. La sous-couche de pré-synchronisation est également responsable de la synchronisation logique inter-flux lorsque des pertes sont provoquées, même si son rôle premier est d'éviter que ces mécanismes de perte n'interviennent pendant l'exécution de l'application. De plus, la sous-couche de pré-synchronisation n'existe que chez le récepteur, et n'a pas son pendant sur la machine émettrice : ce n'est en fait qu'une sous-couche applicative qui permet d'adapter le service transport aux besoins de l'application de synchronisation.

La partie synchronisation terminale de l'application n'est pas modifiée, i.e. tous les mécanismes de contrôle de la synchronisation intra et inter-flux sont maintenus, puisqu'ils correspondent très exactement au comportement modélisé par les TSPN ; les processus de présentation et d'orchestration ne sont pas modifiés. Toutefois, même si la

tâche de synchronisation finale n'est pas modifiée, tout est fait par l'ordre partiel et la pré-synchronisation pour que les mécanismes exceptionnels des processus de présentation et d'orchestration (tels que l'expiration des timers temps réel) n'interviennent pas. Ils ne peuvent cependant pas être supprimés car l'ordre partiel et la pré-synchronisation s'exécutent respectivement dans les classes système et temps partagé du système opératoire, et aucun des services temporels qu'ils essaient d'assurer n'est garanti (à cause des problèmes d'asynchronisme du système opératoire). Par exemple, si un phénomène de gigue important apparaît, suite à une augmentation brutale de la charge machine, la pré-synchronisation risque de ne pas prendre la main avant la date limite, et le temps de présentation maximal sera garanti grâce aux mécanismes temps réel de la partie orchestration de l'application.

Avantages de cette architecture :

- A cause du système asynchrone utilisé, la synchronisation temporelle ne peut être réalisée ailleurs qu'au plus haut niveau de l'architecture de synchronisation, i.e. dans la sous-couche la plus haute de l'application. Cependant, dans cette architecture toutes les opérations qui peuvent être traitées plus tôt, i.e. celles qui ne seront pas annulées lors du passage dans le système opératoire, le sont. Traiter les données au plus tôt est fondamental, car cela permet des optimisations, les données temporellement délicates pouvant être détectées au plus tôt, et les traitements d'exceptions associés pouvant être initiés tant qu'il en est temps. En fait, dans cette architecture, toutes les opérations sont réalisées une fois et une seule, et le plus tôt possible, de façon à garantir une qualité de service maximale.
- Dans cette architecture, les données non synchronisables sont détectées au plus tôt, et sont éliminées au plus tôt pour éviter des pertes de temps qui sont préjudiciables à la qualité de présentation de l'application.
- Dans cette architecture, les stockages sont limités ; seule la sous-couche de pré-synchronisation gère des tampons (si on omet le transport qui ne se situe pas dans le même espace d'adressage).
- Enfin, l'application de synchronisation (présentation et orchestration) ne change pas, sauf par rapport à l'attente d'un message qui peut être raccourcie si le transport ou la pré-synchronisation déclare la donnée invalide temporellement.

4. Description des services fournis par le transport à ordre partiel et la pré-synchronisation

Les services fournis par ces deux couches ou sous-couches ont déjà été abordés succinctement dans ce qui précède, tout au moins dans leur principe. Ce paragraphe présente en détail les différentes primitives de service mises à disposition de l'utilisateur par ces couches et sous-couches, puis présente l'architecture complète des TSPN permettant de modéliser l'application PNSVS 2.

4.1. Services fournis par la pré-synchronisation

La sous-couche de pré-synchronisation, comme cela a déjà été dit, n'apporte qu'un contrôle temporel sur les objets délivrés par le service transport. C'est elle qui stocke les données avant présentation, qui contrôle que le retard de présentation ne dépasse pas la borne autorisée, qui assure la synchronisation logique inter-flux lorsque des pertes sont provoquées, et qui déclare perdues au plus tôt les données attendues qui ne sont plus synchronisables.

De ce fait, et comme la pré-synchronisation n'existe que pour l'application réceptrice, le service fourni par cette sous-couche ne comporte que trois primitives. La première permet à l'utilisateur de la pré-synchronisation de lui communiquer les paramètres de service qu'il souhaite voir respecter (retard de présentation maximal, TSPN de

l'application, etc. ...). Les deux autres primitives sont des indications de la pré-synchronisation vers l'application de synchronisation qui indiquent les données reçues, et les données perdues ou déclarées perdues. Ces deux primitives sont DATA_IND et LOST_DATA_IND respectivement.

4.2. Services fournis par le transport à ordre partiel

Le transport à ordre partiel permet le transfert de bout en bout à haut débit de données qui seront délivrées à l'utilisateur en respectant une qualité de service minimale fixée par l'utilisateur, et une délivrance au plus tôt des données au regard d'un ordre partiel prédéfini. Les primitives de service du transport à ordre partiel sont conformes à celles généralement rencontrées avec les services transport en mode connecté (TCP). Elles permettent l'ouverture de connexions et de multi-connexions à ordre partiel, l'émission et la réception de message. Toutefois, comme le service transport essaie de garantir une certaine qualité de service, les paramètres de QoS sont passés par l'utilisateur lors de la demande d'ouverture de connexion, la requête pouvant être cependant rejetée si la QoS demandée n'est pas garantissable par le service transport. De même, lorsque la connexion est établie, si par moment la QoS demandée par l'utilisateur n'est pas respectée, le service transport le signalera à l'utilisateur par l'intermédiaire d'indications.

La liste des primitives de service n'est actuellement pas figée. Cette liste est présentée dans [CHA95c]. Cependant, l'application PNSVS qui utilise le service transport requiert que les paramètres présentés sur la figure III.4, notamment au moment de l'ouverture de connexion, soient pris en compte par l'ordre partiel.

PRE-SYNCHRONISATION	TRANSPORT A ORDRE PARTIEL
ORDRE réseau de Petri des compositions séries / parallèles	→
FIABILITE nombre de pertes max / s ou / période classe (cf. [CHA95d])	→ →
DONNEES TEMPORELLES -DEBIT flux isochrone (oui / non) tailles des SDU cadence demandée (SDU / s) gigue maximale tolérable sur les objets	→ → → →
	← Délai de transit minimal

Figure III.4. Paramètres de services dans le cas de flux isochrones

Les paramètres de la figure précédente sont classés en 3 groupes : les paramètres liés à l'ordre partiel, à la fiabilité partielle et au temps ou au débit induit.

- Par rapport à l'ordre, le réseau de Petri des compositions séries /parallèles est passé par la pré-synchronisation au transport à ordre partiel pour que ce dernier puisse respecter les principes de fiabilité partielle et de délivrance au plus tôt, requis par l'application. Toutefois, la pré-synchronisation passera un réseau de Petri des compositions séries /parallèles global dans le cas où une multi-connexion multimédia à ordre partiel sera utilisée [CHA95d], ou plusieurs réseaux de Petri dans le cas où plusieurs mono-connexions monomédias à ordre partiel seront utilisées [AME94b].
- Par rapport à la fiabilité, le nombre de pertes par période ou par seconde tolérable par l'application est passé au transport, de même que la classe (définie dans [CHA95c]) qui précise si les pertes sont acceptables et si des mécanismes de recouvrement d'erreur peuvent être mis en place.
- Enfin, par rapport au temps ou au débit induit, l'utilisateur du service transport précise à ce dernier si le flux qu'il souhaite transmettre est isochrone, la taille des SDU, et la cadence normale d'émission et de réception des SDU. Ces paramètres tiennent aussi compte des connaissances dont le transport a besoin pour d'éventuelles réservations de bande passante au niveau réseau.

De plus, la pré-synchronisation précise au service transport la gigue transport maximale (en ms) qui est acceptable sur un objet (cette gigue étant liée au retard maximal autorisé par la pré-synchronisation, et au dessus par l'application de synchronisation), même si à cause de l'asynchronisme système cette gigue maximale ne peut être effectivement garantie. Quoi qu'il en soit, cette information permet au transport de connaître le nombre maximal d'objets qu'il peut stocker dans ses tampons afin de ne pas dépasser la gigue maximale acceptable, et surtout lorsqu'il détecte une perte, il peut déterminer le nombre maximal de retransmissions qui peuvent être effectuées pour recouvrer la perte initiale de cet objet.

Exemple :

Prenons comme exemple pour illustrer le principe précédent le cas du flux vidéo de PNSVS 2, et supposons que l'on veuille faire de la visioconférence à 25 images/s, c'est à dire qu'une donnée vidéo sera produite et émise par l'émetteur toutes les 40 ms ($=\Delta t$) et une de ces données devra être reçue et présentée par le récepteur toutes les 40 ms également. Supposons de plus que le nombre de pertes maximal acceptable est d'une image par seconde, et que la gigue maximale acceptable par la pré-synchronisation est de 100 ms. Soit le scénario de la figure III.5.

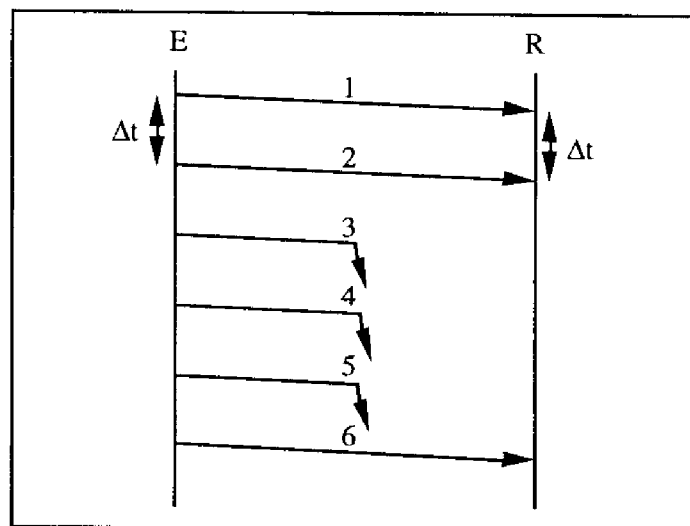


Figure III.5. Principe du contrôle de la gigue maximale du transport à ordre partiel

Sur ce scénario, lorsque le récepteur R reçoit l'image 6, il sait que les données 3, 4 et 5 ont été perdues. 3 pertes par seconde sont interdites, aussi R doit demander à

l'émetteur E des retransmissions des données perdues. Comme le transport ne gère pas le temps de manière explicite, il ne peut pas tenir compte des phénomènes de gigue, et ne peut raisonner que par rapport à des cas idéaux. Ainsi, il doit supposer que toutes les données lui parviennent en un temps égal au délai de transit minimum ($D_{t_{min}}$), et de ce fait, l'écart entre les réceptions de deux images consécutives est égal à Δt . En respectant ce principe, la couche transport de R peut déterminer que :

L'image 1 doit arriver à $D_{t_{min}}=10$ ms et doit être délivrée avant $D_{t_{min}} + \text{Gigue}_{max}= 110$ ms.

L'image 2 doit arriver à $D_{t_{min}} + \Delta t =50$ ms et doit être délivrée avant $D_{t_{min}} + \text{Gigue}_{max} + \Delta t = 150$ ms.

L'image 3 doit arriver à 90 ms et doit être délivrée avant 190 ms.

L'image 4 doit arriver à 130 ms et doit être délivrée avant 230 ms.

L'image 5 doit arriver à 170 ms et doit être délivrée avant 270 ms.

L'image 6 doit arriver à 210 ms et doit être délivrée avant 310 ms.

Lorsque R reçoit l'image 6, il sait qu'il est au moins 210 ms et que les images 3, 4 et 5 ont été perdues. Il est donc trop tard pour l'image 3 qui aurait dû être délivrée avant 190 ms. Par contre, il est encore temps de demander à l'émetteur la réémission des images 4 et 5 (une réémission pouvant ne coûter que $2D_{t_{min}} = 20$ ms).

Le service transport retourne également un paramètre de service à la sous-couche de pré-synchronisation. Ce paramètre est le délai de transit minimal sur la connexion transport, cette information étant nécessaire à la sous-couche de pré-synchronisation dans le cadre du contrôle du retard de bout en bout (cf. formules du chapitre 2 propres à ce problème).

5. Modélisation des niveaux de synchronisation

Dans l'architecture proposée dans ce chapitre, les différentes opérations de synchronisation multimédia ont été réparties en trois couches ou sous-couches :

- le transport à ordre partiel qui optimise les communications par rapport au temps et à la fiabilité ;
- la pré-synchronisation qui délivre les données de manière temporellement optimale à l'application ;
- et l'application qui assure la présentation synchronisée des différents objets en respectant les contraintes de synchronisation définies par le TSPN présentation de l'application.

De plus, ces couches et sous-couches s'occupent également du contrôle du retard de présentation de bout en bout et de la synchronisation logique inter-flux qui peut être perturbée par certains mécanismes applicatifs.

Toutefois, la fonction première de ces couches est avant tout de réaliser une présentation synchronisée de qualité ; ainsi, comme il a été choisi de représenter les contraintes et les mécanismes de synchronisation multimédia par des réseaux de Petri, cette partie présente comment le comportement de chaque couche peut être modélisé par un réseau de Petri, et surtout comment la couche en question peut déterminer ce réseau à partir du réseau de Petri de sa couche cliente. Ainsi, tous les réseaux de Petri peuvent être déterminés à partir du TSPN présentation, qui est lui même déterminé à partir de la QoS définie par l'utilisateur.

5.1. TSPN applicatif

Dans ce paragraphe, le modèle TSPN applicatif défini au chapitre 2 va être repris. En effet, comme de nombreuses fonctionnalités de l'application de synchronisation et de

gestion de la QoS de PNSVS 1 ont été descendues vers le transport et la pré-synchronisation, le nouveau modèle ne doit plus en tenir compte. Le nouveau TSPN applicatif qui modélise les mécanismes de synchronisation de la sous-couche de synchronisation est donné sur la figure III.6 en respectant le même principe de présentation que la figure II.11 ; il ajoute donc aux éléments apparaissant dans le TSPN de présentation tous les paramètres associés aux places qui permettent de mettre en oeuvre et de configurer les algorithmes de synchronisation intra et inter-flux. De plus, d'autres paramètres concernant la fiabilité, et donc n'ayant aucun lien avec le TSPN applicatif, sont également donnés. Ces paramètres concernent pour la sous-couche de synchronisation la qualité de présentation minimale exigée par l'utilisateur, et définissent en particulier le nombre maximal de discontinuités autorisées.

	paramètres fournis par l'utilisateur	paramètres mesurés par l'application	paramètres déterminés par l'application
TSPN applicatif émetteur	type de synchronisation		Réseau de Petri du comportement applicatif de l'émetteur $[t_{\min}, t_{\text{nom}}, t_{\max}]$ $[t'_{\min}, t'_{\text{nom}}, t'_{\max}]$
TSPN applicatif récepteur	type de synchronisation	t_{l_r} t'_{l_r} τ_{RT} $t_{\min}(\text{traitement1})$ $t_{\min}(\text{traitement2})$ $t_{\min}(\text{traitement3})$ $t_{\min}(\text{décomp/affich})$	Réseau de Petri du comportement applicatif du récepteur $[t_{\min}, t_{\text{nom}}, t_{\max}]$ $[t'_{\min}, t'_{\text{nom}}, t'_{\max}]$
paramètres de fiabilité pour le récepteur	nombre maximal de discontinuités par seconde		

Figure III.6. Définition des TSPN applicatifs et des paramètres de services associés à la gestion de la QoS

5.2. TSPN de pré-synchronisation

La pré-synchronisation a pour rôle de préparer les données qui lui sont transmises par le transport, et de les passer de façon temporellement optimale à l'application qui doit les présenter. Cette sous-couche compense les problèmes temporels de la couche transport et réalise en fait un réordonnement temporel des messages reçus, de telle façon que l'application de synchronisation ait l'impression que le support de communication qu'elle utilise est faiblement synchrone (faiblement synchrone à cause de l'asynchronisme du système opératoire en classe temps partagé). Ces fonctionnalités de la pré-synchronisation sont donc modélisées par un TSPN de pré-synchronisation permettant de décrire le comportement de cette sous-couche, laquelle permet d'améliorer la qualité temporelle des communications au regard des besoins de l'application. Le TSPN de pré-synchronisation est représenté sur la figure III.7. Concernant cette fonctionnalité, le temps maximal d'attente d'un objet est associé à la place correspondante ; ce paramètre est le seul qui soit nécessaire à la mise en oeuvre

de la pré-synchronisation temporelle. De plus, comme la pré-synchronisation est également chargée de contrôler le retard de bout en bout, ainsi que la synchronisation logique inter-flux lorsque des pertes sont provoquées, la figure III.7 associe tous les paramètres de gestion du retard de bout en bout aux places correspondantes (cf. les formules de dimensionnement des tampons de réception vues au chapitre 2).

	paramètres fournis par l'utilisateur	paramètres mesurés par l'application	paramètres déterminés par l'application
TSPN de communication			Réseau de Petri extrait du TSPN applicatif du récepteur t_{\max} t'_{\max}
paramètres de contrôle du retard	retard de présentation maximal autorisé	t_{l_e} t'_{l_e} $t_{\min}(\text{trait}_e)$ $t'_{\min}(\text{trait}_e)$ $t_{\min}(\text{trait}_r)$ $t'_{\min}(\text{trait}_r)$ t_{l_r} t'_{l_r} t_{nom} t'_{nom}	$d_{\text{com}_{\min}}$ $d'_{\text{com}_{\min}}$

Figure III.7. Définition du TSPN de communication et des paramètres de contrôle du retard

5.3. Réseau de Petri des compositions série / parallèle

Le transport à ordre partiel réalise le transfert à haut débit des données, et ceci grâce aux mécanismes de fiabilité partielle sur une ou plusieurs connexions. Cette fiabilité partielle est basée sur un ordre partiel des objets à délivrer, et cet ordonnancement est représenté par un réseau de Petri des compositions série / parallèle des divers objets dans le cas d'une multi-connexion multimédia à ordre partiel [DIA94c] ou par deux réseaux de Petri (un pour la connexion audio et un pour la connexion vidéo) si deux mono-connexions monomédias à ordre partiel sont utilisées [AME94b]. Par rapport aux autres mécanismes de la couche transport, à savoir une gestion de la fiabilité par média et par groupe de médias, et le contrôle du délai de transit, les paramètres déjà évoqués, comme la cadence en terme de SDU par seconde, la gigue maximale autorisée, la classe de fiabilité et le nombre de pertes maximal par seconde sont nécessaires.

5.3. Exemple complet

Cette partie dresse un récapitulatif qui présente la hiérarchie des TSPN qui permettent de modéliser le comportement de chacune des couches et sous-couches de l'architecture de PNSVS 2. Ainsi, en partant du TSPN de présentation de la figure I.9, la figure III.8 montre les différentes déductions qui permettent d'obtenir les TSPN applicatifs, le TSPN de pré-synchronisation (modélisant la pré-synchronisation), et le réseau de Petri des compositions série / parallèle pris en compte par le transport à ordre

partiel. Cependant, tous les paramètres des différents TSPN n'ont pas été mentionnés sur la figure III.8 pour des raisons de clarté.

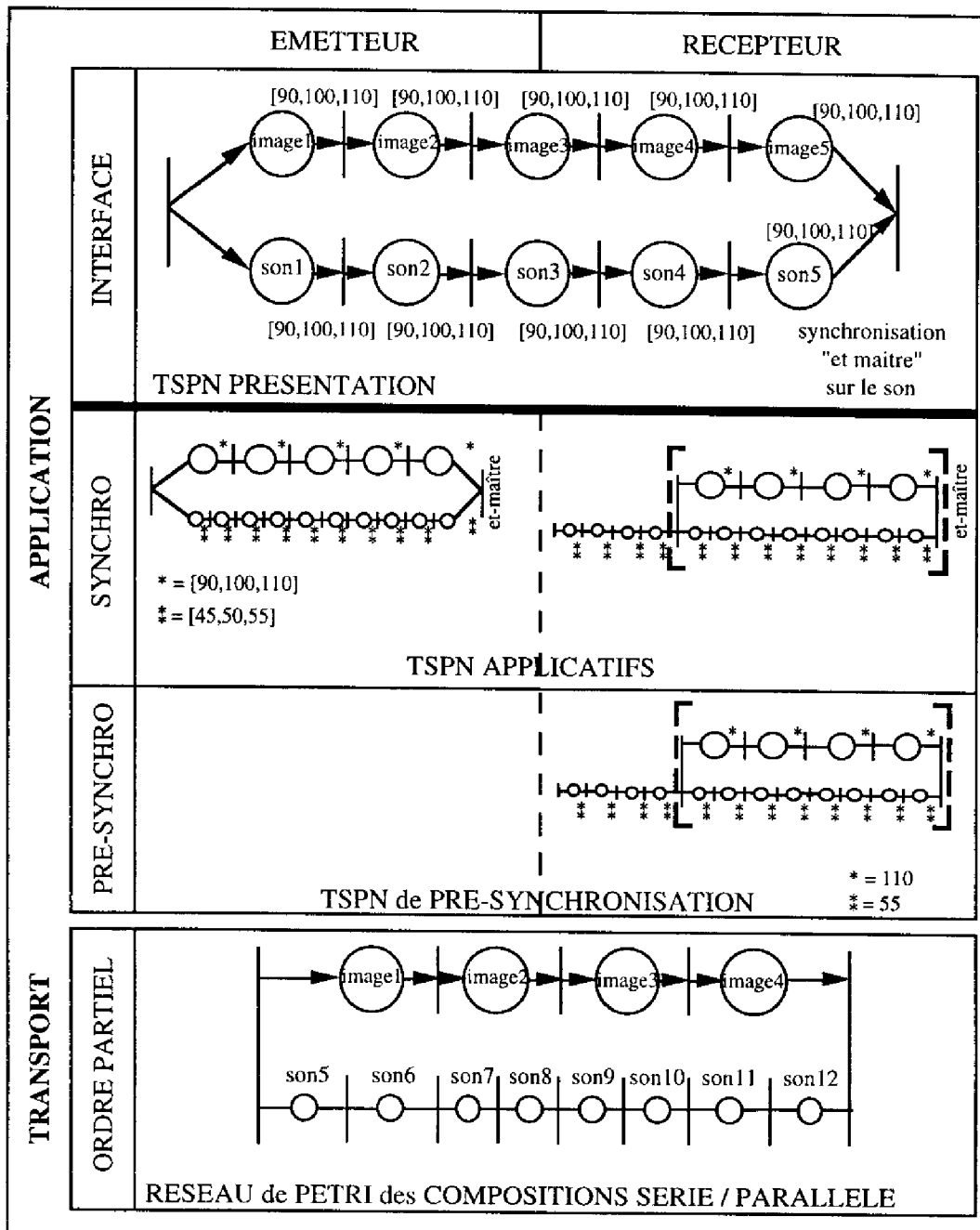


Figure III.8. Architecture des TSPN modélisant PNSVS 2

6. Implémentation

Les services fournis par les couches et sous-couches transport et pré-synchronisation étant définis, cette partie va présenter les algorithmes des différents composants de l'architecture de synchronisation de PNSVS 2, représentés sur la figure III.9. Comme

pour le chapitre 2, seuls les algorithmes pour la partie vidéo seront présentés, ceux concernant les données audio étant similaires.

6.1. Architecture et algorithmes de l'application de synchronisation

L'architecture des threads mise en oeuvre dans PNSVS 2 ressemble beaucoup à celle de PNSVS 1, avec les threads de présentation, de stockage et d'orchestration pour chacun des flux audio et vidéo. La différence provient de la tâche de pré-synchronisation ; cette tâche a pour rôle d'assurer un contrôle temporel sur l'activité du protocole de transport. Elle peut ainsi être rapprochée des orchestrateurs : comme les orchestrateurs qui contrôlent temporellement les processus de présentation, la pré-synchronisation contrôle temporellement les communications. Finalement, par rapport à PNSVS 1 qui n'intègre qu'un seul niveau de synchronisation - le niveau présentation / orchestration - , PNSVS 2 intègre un second niveau de synchronisation : le niveau communication / pré-synchronisation (figure III.9).

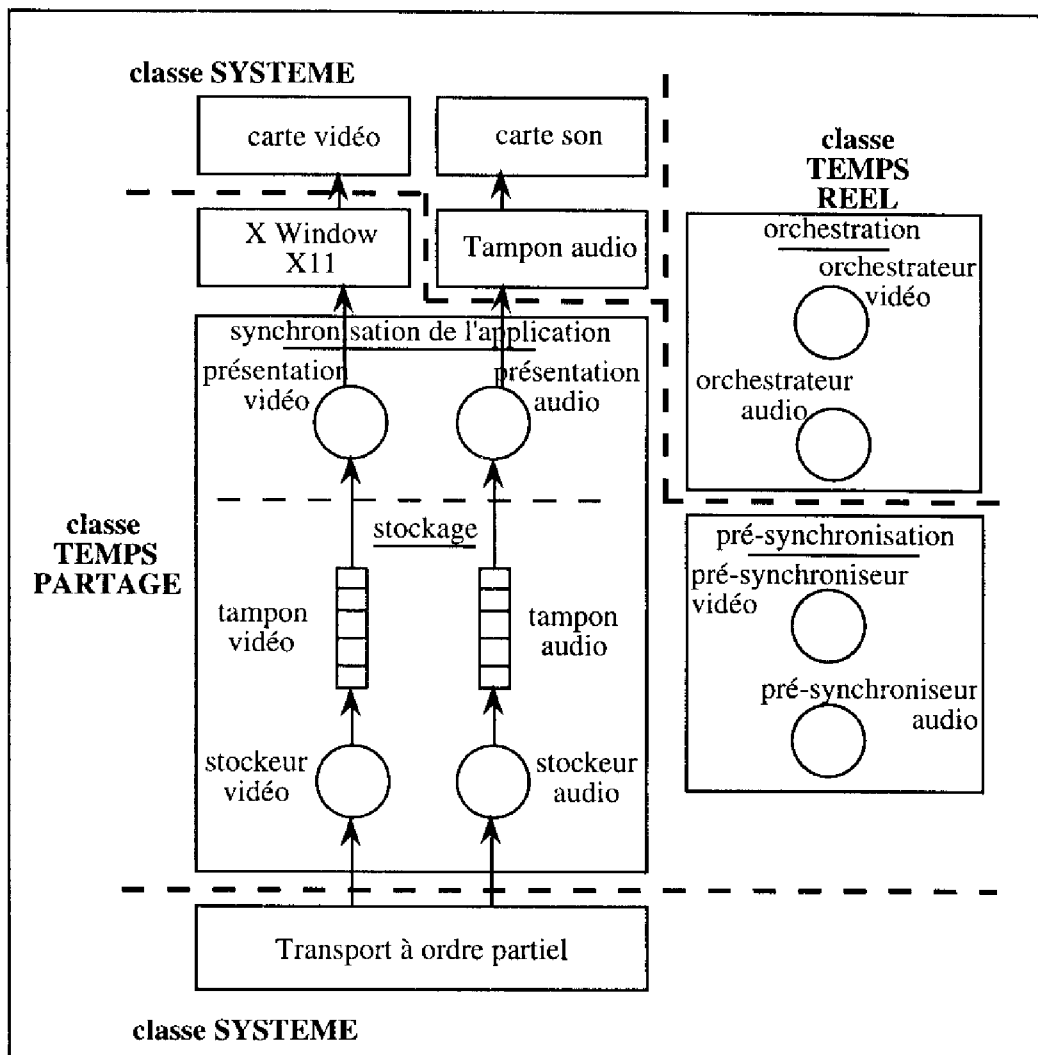


Figure III.7. Architecture de synchronisation pour PNSVS 2 au dessus d'un transport à ordre partiel

L'application de synchronisation finale est à un détail près la même que celle utilisant un transport classique style UDP. La seule différence lorsque l'architecture complète de PNSVS 2 (transport à ordre partiel, pré-synchronisation, synchronisation) est utilisée concerne la possibilité pour la sous-couche de synchronisation de bénéficier d'une information supplémentaire en provenance du transport ou de la pré-synchronisation, signalant au plus tôt les données perdues ou dont la gigue ne sera pas rattrapable. Ceci rajoute donc une autre valeur au niveau de la variable statut qui donne l'état de la donnée contenue dans le tampon : en plus des valeurs "donnée présente et donnée non arrivée" vues au chapitre 2, la donnée en question peut avoir le statut de "donnée perdue".

En ce qui concerne les variables de synchronisation entre les threads de PNSVS 2, les mutex et variables conditions restent utilisées comme cela a été expliqué dans le chapitre 2. Toutefois, le tableau de variables condition cv disparaît et est remplacé par deux autres tableaux d'éléments du même type : cv_s fait le lien entre la présentation et la pré-synchronisation ; cv_ps fait le lien entre la pré-synchronisation et le stockage. La figure III.10 représente l'architecture des variables de synchronisation pour la partie vidéo de PNSVS 2.

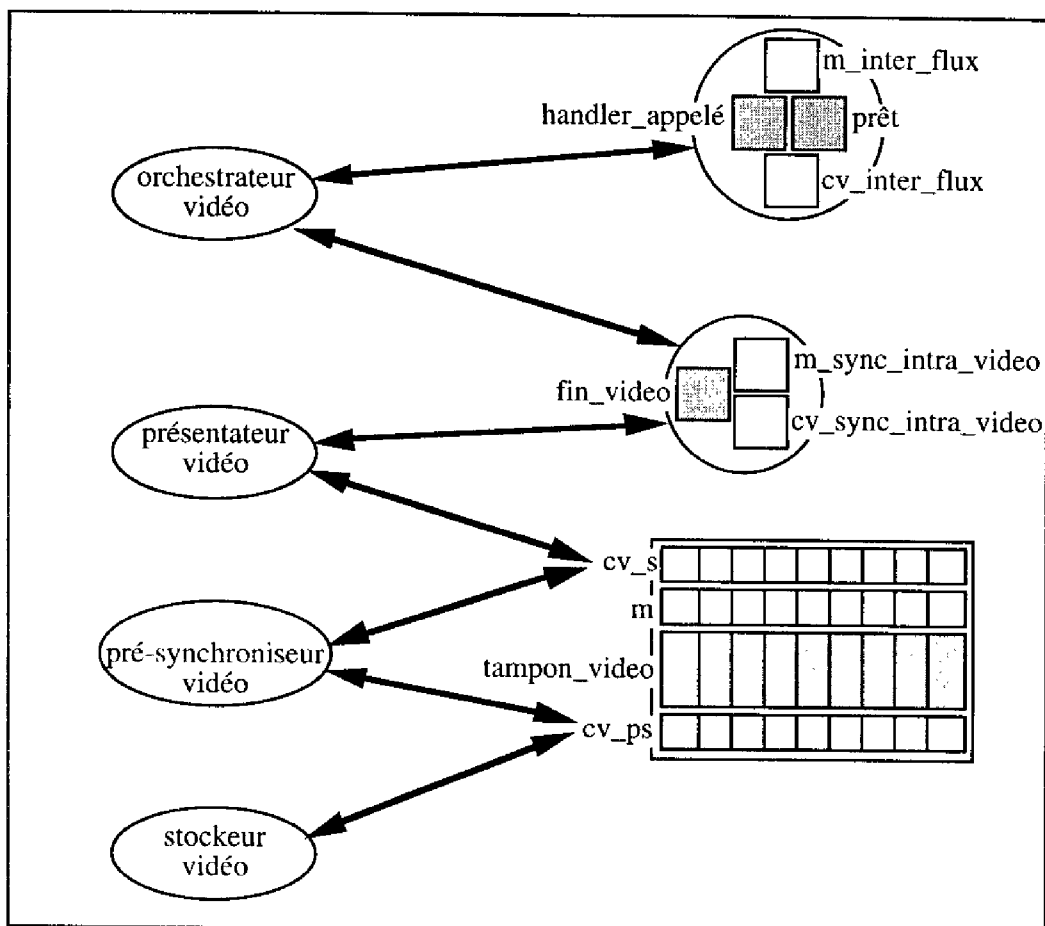


Figure III.10. Architecture des variables de synchronisation de la partie vidéo de PNSVS 2

Algorithme du thread de présentation vidéo

```
fin_vidéo ← Faux ;
```

```
/* présenter l'image courante */
```

```

verrouiller le mutex de m associé à l'image courante ;
Si le statut de l'image courante est "image présente" alors
  /* cas normal de présentation d'une image */
  repositionner le statut de l'image à "image non arrivée" ; /* ré-initialisation pour le prochain
  passage */
  signaler à la pré-synchronisation l'instant à partir duquel elle doit contrôler le  $t_{max}$ . Pour
  cela, il faut libérer le thread de pré-synchronisation qui attend sur la variable condition cv_s ;
  libérer le mutex de m associé à l'image courante ;
  décompresser et afficher l'image courante ;
  attendre  $t_{min} - t_{min}(\text{décompression/affichage}) - t_{min}(\text{traitement}_1)$  ;
sinon
  /* exception : l'image précédente va remplacer l'image courante car le statut de l'image est
  soit "donnée non arrivée", soit "donnée perdue" */
  repositionner le statut de l'image à "donnée non arrivée" ; /* ré-initialisation pour le prochain
  passage */
  signaler à la pré-synchronisation l'instant à partir duquel elle doit contrôler le  $t_{max}$ . Pour
  cela, il faut libérer le thread de pré-synchronisation qui attend sur la variable condition cv_s ;
  libérer le mutex de m associé à l'image courante ;
  attendre  $t_{min} - t_{min}(\text{traitement}_1)$  ;
fin si ;

/* attendre l'image suivante */
verrouiller le mutex de m associé à l'image suivante de l'image courante ;
si le statut de cette future image est "image non arrivée" alors
  bloquer le processus de présentation vidéo sur la variable condition de cv_s associée à la
  prochaine image ; /* il sera libéré soit par le processus d'orchestration vidéo si le timer
  temps réel expire, soit par le processus de stockage vidéo si il reçoit l'image attendue à
  temps, soit par la pré-synchronisation si la donnée n'arrive pas à temps */
fin si ;

libérer le mutex de m associé à l'image suivante de l'image courante ;

/* signaler la fin de la présentation à l'orchestrateur vidéo */
verrouiller le mutex m_sync_intra_vidéo ;
fin_vidéo ← Vrai ;
libérer le processus d'orchestration vidéo bloqué sur la variable condition cv_sync_intra_vidéo ;
libérer le mutex m_sync_intra_vidéo ;

terminer le processus de présentation vidéo ;

```

Orchestration

En ce qui concerne les processus d'orchestration, tous les mécanismes qui permettent de jouer un TSPN et de respecter les contraintes temporelles de présentation doivent être maintenus, car la pré-synchronisation s'exécutant en classe temps partagé subit l'asynchronisme du système opératoire et ne fournit donc pas un service garanti. L'algorithme du processus d'orchestration vidéo est donc toujours le même.

6.2. Pré-synchronisation

Algorithmes des threads de stockage

```

Répéter à l'infini :
  réceptionner un message ;
  sauvegarder le numéro de séquence de ce message ;
  stocker le message dans le tampon vidéo en fonction de son numéro de séquence ;

```

```

pour tous les messages dont le numéro de séquence est compris entre le dernier reçu et
celui qui vient d'être reçu :
    positionner le statut de ce message à "donnée perdue" ; /* garanti de par le service
d'ordonnement du transport à ordre partiel */
    signaler à la pré-synchronisation cette perte en la libérant éventuellement de la variable
condition cv_ps ;
fin de boucle pour ;

/* prévenir la pré-synchronisation de l'arrivée d'une donnée */
verrouiller le mutex de m associé au numéro de séquence du message reçu ;
positionner le statut de ce message à "donnée présente" ;
signaler à la pré-synchronisation l'arrivée de cette donnée. Ce signal est envoyé sur la
variable condition de cv_ps associée au numéro de séquence du message reçu, et peut
éventuellement libérer le processus de pré-synchronisation vidéo ;
libérer le mutex de m associé au numéro de séquence du message reçu ;
fin de boucle répéter

```

Algorithme des threads de pré-synchronisation

```

verrouiller le mutex de m associé à l'image courante ;
se bloquer sur la variable condition cv_s associée à l'image courante pour attendre le signal de
contrôle du  $t_{max}$  provenant de la présentation vidéo ;
libérer le mutex de m associé à l'image courante ;

sauvegarder l'heure de début du contrôle du  $t_{max}$  dans heure_sauv;

Répéter à l'infini :
    verrouiller le mutex de m associé à l'image suivant l'image courante ;
    si le statut de cette future image est "donnée non arrivée" alors
        calculer le temps restant à attendre au maximum
            (temps restant =  $t_{max}$  - (heure actuelle - heure_sauv) ) ;
        se bloquer sur la variable condition cv_ps associée à l'image attendue et attendre le
signal du thread de stockage vidéo. Au cas où ce signal ne viendrait pas à temps, armer
le temporisateur associé à la variable condition ;
        si le temporisateur a expiré alors
            positionner le statut de l'image attendue à "donnée perdue" ; /* cette donnée est trop
en retard pour pouvoir être présentée */
        fin si ;
    fin si ;

    signaler au thread de présentation que la pré-synchronisation a traité la prochaine image à
présenter. Pour cela, envoyer le signal sur la variable condition cv_s pour éventuellement
libérer le thread de présentation ;

    se bloquer sur la variable condition cv_s associée à l'image courante pour attendre le signal
de contrôle du  $t_{max}$  provenant de la présentation vidéo ;

    libérer le mutex m associé à la prochaine image à présenter;

    sauvegarder l'heure de début du contrôle du  $t_{max}$  dans heure_sauv;
fin de boucle répéter ;

```

Remarque :

Les algorithmes précédents, en particulier ceux de la pré-synchronisation, ne tiennent pas compte de l'ordonnement logique inter-flux des objets que respecte le transport à ordre partiel dans ses délivrances. En effet, à cause du système opératoire asynchrone, l'ordonnement inter-flux des objets peut être modifié. De même, si une multi-connexion multimédia à ordre partiel est utilisée, à cause du non respect de l'ordonnement transport par le système, les pré-synchroniseurs, lorsqu'ils reçoivent

un objet, ne tirent pas de conclusion quant à la perte de données sur d'autres flux. En fait, seules les contraintes d'ordre et de fiabilité (perte et délivrance au plus tôt) sur un flux sont garanties, et la sous-couche de pré-synchronisation fonctionne comme si elle utilisait deux mono-connexions à ordre partiel.

6.3. Transport à ordre partiel

Les principes importants pour PNSVS 2 de l'ordre partiel ont été décrits dans les parties précédentes. Toutefois, PNSVS 2 ne fait qu'utiliser ce service transport, et les algorithmes qui permettent d'implanter ces principes ne sont donc pas du ressort de cette thèse. Pour cela, le lecteur pourra se rapporter à [DIA94c][AME94b][CHA95a][CHA95d].

Conclusion

Le but de ce chapitre consistait à analyser le problème de dégradation importante de la qualité de service que l'on observait lorsqu'une perte au niveau du support de communication se produisait avec l'application de visioconférence décrite dans le chapitre 2, puis à essayer de le résoudre. En fait, ce problème est lié aux attentes de l'application en cas de problèmes sur le support de communication (gigue importante ou perte d'une donnée), et les mécanismes de contrôle des retards et d'accélération de flux provoquaient ces trop nombreuses pertes. Aussi, il est apparu comme essentiel de disposer d'un support de communication capable de détecter les pertes au plus tôt, et de les signaler au plus tôt à l'utilisateur afin que ce dernier puisse réagir le plus rapidement possible. Ainsi, un transport à ordre partiel a été utilisé, et cette partie décrit l'architecture qu'il a fallu mettre en place. Ensuite, les algorithmes de chacune des couches et sous-couches de l'architecture (ordre partiel, pré-synchronisation et synchronisation temporelle) ont été donnés, ainsi que les TSPN qui modélisent leurs comportements.

Avec l'architecture et les mécanismes présentés dans cette thèse, on obtient un système de visioconférence extrêmement performant au niveau de la qualité de service et de la synchronisation obtenues (cf. l'évaluation des résultats obtenus est présentée dans [BOY96b]). PNSVS 2 répond donc parfaitement à la première partie de la problématique exposée au chapitre 1. Il ne reste donc plus qu'à étudier et à intégrer dans une application de visioconférence des mécanismes de coopération pour répondre à la double problématique posée dans le cadre de cette thèse. L'aspect coopératif pour N utilisateurs est abordé au chapitre suivant.

Chapitre IV

Conception et implémentation d'un système de visioconférence coopératif

Introduction

Le chapitre 1 a montré que de nombreuses applications de visioconférence ont vu le jour. Cependant, aucune de ces applications n'a été conçue pour supporter un travail de groupe structuré ayant un fort degré d'interactivité entre les participants. En effet, aucun de ces systèmes n'intègre formellement la notion de groupe coopératif, ni même la notion de gestion de la qualité de service dans un groupe. Ce sont ces problèmes que le prototype N-TSVS tente de résoudre. Tout d'abord, le travail coopératif est une activité qui doit être structurée entre des participants prédéfinis et qui travaillent en respectant des règles bien déterminées dépendant du type d'activité du groupe. Aussi, N-TSVS intègre des fonctionnalités de gestion de groupes dynamiques en contrôlant les entrées et sorties des participants à la visioconférence, de telle sorte que les règles régissant l'activité du groupe soient toujours respectées. De même, N-TSVS contrôle toutes les interactions entre les différents membres du groupe de façon à ce que leur travail respecte les règles qui ont été établies pour la session de travail. De plus, comme une application de visioconférence est avant tout une application multimédia distribuée entre des gens qui disposent de machines et de réseaux de communication dont les possibilités sont différentes, N-TSVS gère au niveau du groupe de travail la qualité de service pour la présentation des objets multimédias et pour les communications. En fait, par la présence conjointe de mécanismes de coopération, de communication de groupe et de synchronisation, N-TSVS peut être qualifiée d'application de visioconférence multipoints à multipoints.

Comme pour TSVS, la synchronisation dans N-TSVS est obtenue grâce à une méthode basée sur l'utilisation d'estampilles. Certes, nous avons vu au chapitre 1 que cette technique ne permettait pas de garantir les contraintes de synchronisation des médias audio et vidéo. Toutefois, dans des conditions favorables de charge machine et réseau, les résultats obtenus sont tout à fait acceptables. De plus, choisir cette approche nous permettait de ne pas mélanger les problèmes de synchronisation avec les problèmes de coopération ; l'objectif premier de ce chapitre est de concevoir et développer des mécanismes de coopération, et dans un premier temps nos efforts se sont portés sur cette partie ; ce choix des estampilles s'est donc imposé par souci de

simplification de la complexité de cette tâche. Néanmoins, une fois les problèmes de coopération résolus, les mécanismes de synchronisation multimédia implémentés dans PNSVS 1 ont été conceptuellement intégrés à cette application de visioconférence coopérative (cf. partie 3.4). Cette intégration ne présente d'ailleurs aucune difficulté conceptuelle.

La suite de ce chapitre respectera le plan suivant : tout d'abord, les principes de conception de N-TSVS seront introduits (partie 1) en présentant et en justifiant la séparation qui a été faite entre d'une part un service de gestion des entrées et sorties dynamiques du groupe de coopération, et une partie traitement des données de l'application de visioconférence avec gestion de la qualité de service d'autre part. Puis la partie 2 présentera plus en détail la partie concernant la gestion des groupes, le service offert et le protocole utilisé, alors que la section 3 développera la partie communication et présentation des données de la visioconférence, la gestion de la qualité de service et de la synchronisation multimédia. La partie 4 montrera comment les erreurs de N-TSVS peuvent être récupérées pour éviter une fin prématurée de l'application, puis la partie 5 exposera une solution au problème de la structuration des communications de la visioconférence. Enfin, en conclusion nous reviendrons sur certains aspects critiques de N-TSVS, et nous conclurons ce chapitre avec quelques perspectives d'évolution pour cette application.

1. Principes de N-TSVS

Lorsque l'on considère les applications de visioconférence, ou plus généralement les applications de travail coopératif, il apparaît toujours une structure du groupe de travail avec des règles qui régissent les interactions entre ses membres (cf. chapitre 1). De plus, de nombreux autres points communs existent, comme la nécessité de pouvoir autoriser des entrées et sorties du groupe à tout moment (dynamisme du groupe) si les règles de la coopération l'autorisent. Aussi, un service générique de gestion des groupes et de la coopération peut intervenir dans de nombreuses applications coopératives. L'application N-TSVS a donc été découpée en deux parties : une partie qui s'occupe de la gestion du groupe des participants, et une partie propre à la visioconférence qui concerne la manipulation, la communication et le traitement des données audio et vidéo de cette application.

La partie gestion des groupes et de la coopération est donc chargée de gérer les groupes, i.e. les entrées et sorties dynamiques du groupe. Le modèle de coopération, présenté au chapitre 1 définit d'abord la structure conceptuelle du groupe [DIA92] [DIA93d] [VIL95]. Cette structure se présente sous la forme d'une activité complexe, formée d'un ensemble d'agents, de données et d'informations que ces agents possèdent, peuvent connaître ou communiquer à d'autres agents, ainsi que des relations possibles entre ces agents. Comme vu au chapitre 1, chaque agent possède un ensemble de données dont il est propriétaire et que, seul, il peut modifier. Lorsqu'un agent change une de ses données, il communique sa nouvelle valeur aux autres agents en relation avec lui, à l'intérieur du groupe. La structure du groupe coopératif donne les relations entre les membres du groupe. Cette structure forme un graphe, le graphe de coopération, dont les sommets représentent les agents, les flèches donnant les relations entre eux. L'approche retenue pour les échanges d'information à l'intérieur des groupes est celle du partage d'information entre agents [DIA92], [KAP92].

Le modèle de coopération donne la structuration globale du groupe d'agents. Par exemple, la figure IV.1 montre la structuration simple d'une visioconférence totale (ou ouverte) pour une réunion entre 4 individus : le graphe résultant est totalement maillé et tous les participants ont le même rôle, car le but est de privilégier l'échange des idées et l'interactivité entre tous les participants. Cependant, pour commencer ou pour réaliser une application de groupe, il n'est pas forcément nécessaire que l'ensemble des agents soient présents en même temps dans le groupe. Le travail coopératif peut donc débiter dès qu'un ensemble adéquat d'agents est actif dans le groupe. Par exemple, dans le cas

de la visioconférence cité précédemment, la présence de deux participants peut être suffisante. Par conséquent, le modèle conceptuel a été étendu (cf. chapitre 1) pour savoir quels agents doivent être présents ensemble pour réaliser l'application coopérative. L'application associée au groupe d'agents doit définir les sous-ensembles d'agents, ou configurations valides, qui ont une signification pour la réalisation effective du travail.

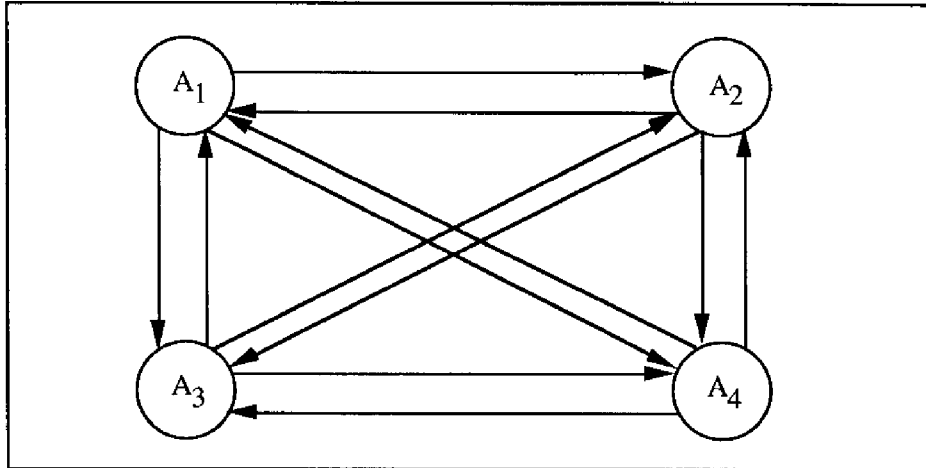


Figure IV.1. Graphe de coopération d'une visioconférence totale

La partie propre à la visioconférence est, quant à elle, chargée de toutes les opérations qui concernent les données audio et vidéo, c'est à dire qu'elle s'occupe de l'acquisition, de la compression, de la transmission, de la décompression et de la présentation des données multimédias, et ce dans des communications multipoints à multipoints. De plus, elle participe à la gestion de la qualité de service des données multimédias pour tout le groupe des participants à la visioconférence. La notion de qualité de service à N est extrêmement complexe, puisqu'elle englobe la notion de qualité de service point à point et doit l'étendre à un groupe d'utilisateurs en tenant compte de paramètres de qualité de service de groupe. Aussi, dans les études que nous avons menées, la définition de la qualité de service à N que nous pouvons faire n'est pas figée, et reste adaptable et modulable. De ce fait, dans la réalisation de N-TSVS, nous avons simplifié à l'extrême cette notion, en ne considérant qu'un seul paramètre de qualité qui est la capacité pour un participant de traiter ou non de la vidéo animée temps réel, et dans l'affirmative, le nombre d'images par seconde qu'il lui est possible de traiter. La partie données de N-TSVS gère aussi la synchronisation multimédia. Cette synchronisation multimédia est difficile à obtenir dans le cadre de visioconférences point à point, mais elle l'est bien plus dans le cadre de N-TSVS, car il faut en plus de la synchronisation des flux en provenance d'un même utilisateur, synchroniser entre eux tous les flux de tous les participants. En ce qui concerne les communications audio et vidéo, des bibliothèques de communication point à multipoints au dessus d'IP multicast ont été réalisées.

La figure IV.2 représente l'architecture de N-TSVS. Elle montre l'agencement des modules pour les sites des participants, avec les deux parties de gestion de la coopération et de traitement des données de la visioconférence ; ces deux parties sont interconnectées par un tube de communication. La figure IV.2 montre aussi l'architecture du site sur lequel s'exécute le gestionnaire de la partie contrôle de la coopération [OWE96a][OWE96b].

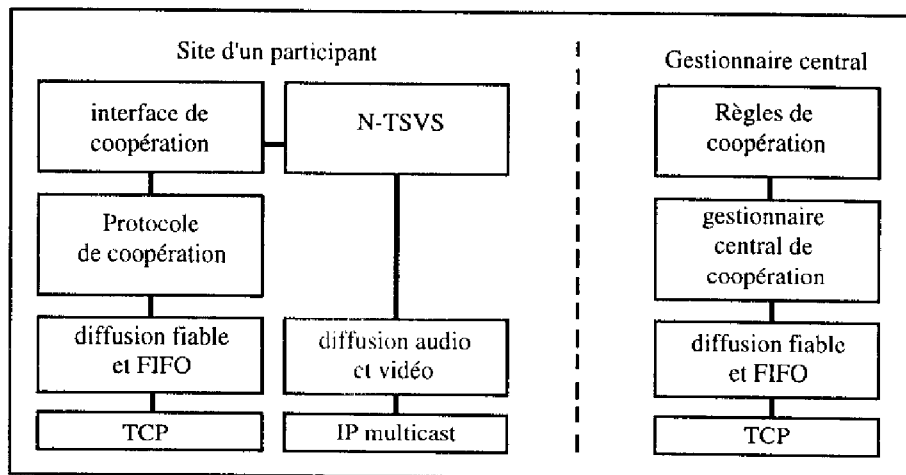


Figure IV.2. Architecture logicielle de N-TSVS

2. La gestion de la coopération dans l'application N-TSVS

- Les deux fonctions clés de N-TSVS consistent à :
- gérer l'activité du groupe coopératif,
 - et traiter les données audio et vidéo de communication.

Cette partie présente le premier point, c'est-à-dire les différents mécanismes et services rendus par chacun des modules de la figure IV.2, ainsi que les protocoles qui permettent à ces entités de gérer le travail coopératif. Le service offert par le protocole de coopération sera décrit, ainsi que le principe de fonctionnement de ce protocole qui, par l'intermédiaire d'un gestionnaire central, permet de gérer les entrées et sorties des participants, la QoS, etc. De même, le fonctionnement de l'interface de coopération sera exposé, puis il sera montré comment l'utilisateur final et la partie traitement des données de N-TSVS utilisent cette interface.

2.1. Le protocole et le service de coopération

2.1.1. Description du service

Le service de coopération proposé gère la dynamique du groupe [DIA93d] et permet le passage d'une configuration valide dans laquelle des agents échangent des informations vers une autre configuration, en considérant les demandes de participation ou d'abandon de l'application de groupe provenant des agents. Supposons d'une part qu'un ensemble d'agents présents dans le groupe, communiquent entre eux des données, et constituent une configuration valide, et d'autre part que d'autres agents non actifs, mais faisant partie du groupe, veulent participer au travail de groupe, et rejoindre ceux qui travaillent déjà. De la même façon, des agents actifs peuvent souhaiter abandonner le travail de groupe, et cesser leur participation au travail coopératif. En considérant les requêtes pour participer ou pour abandonner le travail de groupe, le service essaie de former une nouvelle configuration valide.

Plusieurs choix sont possibles pour changer de configuration valide. Le premier est de changer de configuration dès que possible. Le système qui gère le groupe effectue le changement dès qu'une nouvelle configuration est valide. Le second choix qui a été retenu ici est de considérer que la décision de changer de configuration est une décision coopérative qui concerne l'ensemble des agents réalisant actuellement le travail de

groupe. Par conséquent, la possibilité de changer de configuration valide est proposée à l'ensemble des agents actifs, présents en coopération. La décision finale est prise au moyen d'un vote. Chaque agent en coopération vote pour accepter ou refuser le changement (figure IV.3).

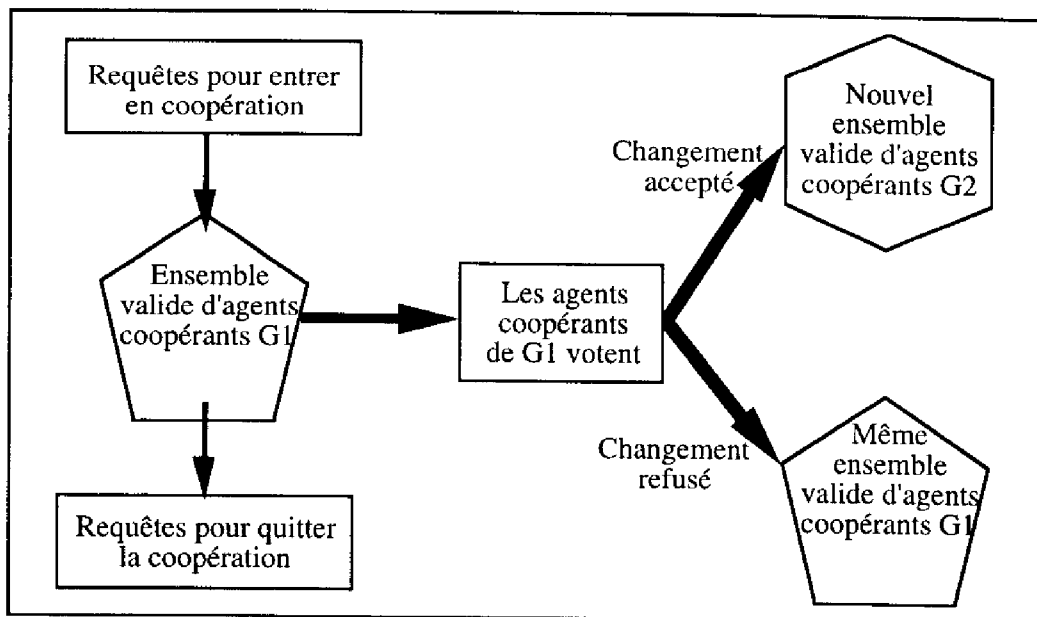


Figure IV.3. Principe de changement de coopération

2.1.2. Fonctionnement du service

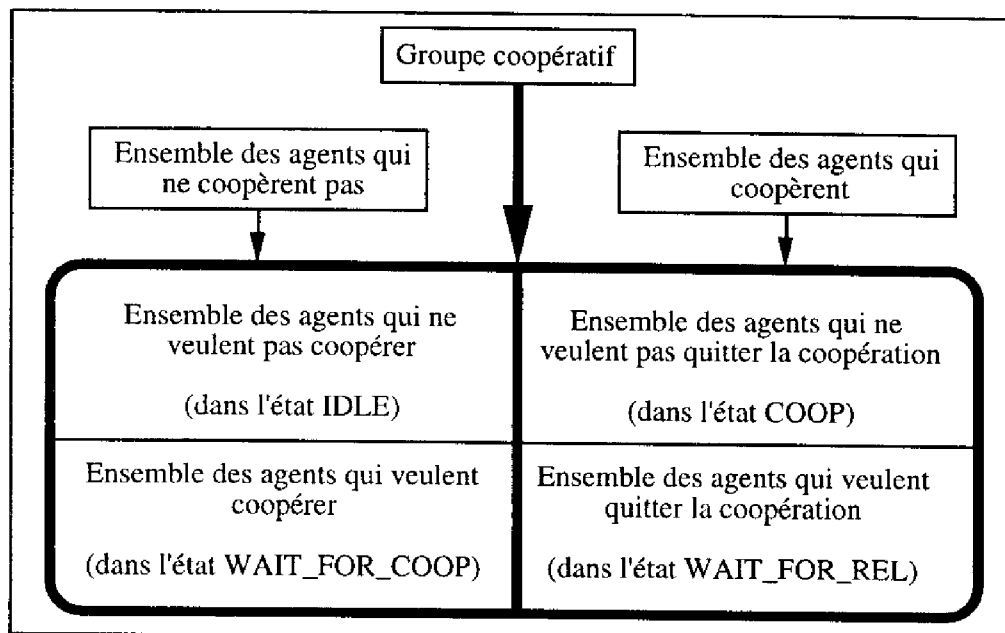


Figure IV.4. Etat de tous les agents du groupe coopératif

Pour décrire le comportement des entités coopérantes, quatre états peuvent être définis pour chacun des agents du groupe coopératif :

- IDLE : l'agent ne coopère pas.

- WAIT_FOR_COOP : l'agent désire participer au travail coopératif.
- COOP : l'agent participe au travail coopératif et ne veut pas abandonner son travail.
- WAIT_FOR_REL : l'agent participe actuellement au travail coopératif, mais il souhaite abandonner le travail dès que possible.

A chaque instant, le graphe instantané de domaine valide est formé des agents dans l'état COOP et des agents dans l'état WAIT_FOR_REL ; les agents ne faisant pas partie de la configuration courante sont soit dans l'état IDLE soit dans l'état WAIT_FOR_COOP (figure IV.4).

Parmi le groupe d'agents coopératifs, le service regarde s'il est possible de former un nouveau graphe instantané de domaine valide : pour cela, il ajoute à la coopération des agents dans l'état WAIT_FOR_COOP, garde tous les agents dans l'état COOP, enlève des agents dans l'état WAIT_FOR_REL et vérifie les règles d'obtention d'un nouveau graphe.

Si aucun nouveau graphe instantané de domaine valide ne peut être formé en ajoutant des agents dans l'état WAIT_FOR_COOP ou en enlevant des agents dans l'état WAIT_FOR_REL, alors le service attend que d'autres agents modifient leur état pour essayer d'obtenir une nouvelle configuration valide. Les sites ayant demandé à entrer en coopération sont alors bloqués en attente d'autres modifications. Les sites ayant demandé à sortir sont également en attente d'autres modifications, mais ils continuent le travail coopératif sans être bloqués.

Supposons cependant qu'un nouveau graphe instantané soit possible. Dans ce cas, l'ensemble des agents coopérants est appelé à voter pour savoir s'ils acceptent le changement ou non. Si ce changement est accepté, alors les agents du nouveau graphe en attente de coopération entrent dans cette dernière (de l'état WAIT_FOR_COOP, ils passent dans l'état COOP) et les agents qui voulaient quitter le groupe et qui sont exclus du nouveau graphe quittent la coopération (ils passent de l'état WAIT_FOR_REL dans l'état IDLE). Après le vote, si le changement est refusé, la structure courante de coopération reste la même. Dans ce cas, les agents qui désirent quitter la coopération et qui n'appartenaient pas au nouveau graphe sont obligés de rester dans le groupe (de l'état WAIT_FOR_REL ils repassent à l'état COOP) et les agents du nouveau graphe qui désiraient entrer en coopération se voient refuser leur entrée (de l'état WAIT_FOR_COOP ils repassent à l'état IDLE).

Le service pour changer de configuration valide nécessite quatre phases différentes pour gérer les interférences entre les échanges de données et le changement de structure. Lorsqu'un changement de configuration a été décidé, les agents de la nouvelle configuration valide doivent échanger leurs contextes initiaux en suivant la structure de la coopération. La seconde phase est celle durant laquelle le travail de groupe est réalisé, i.e. la phase pendant laquelle les agents échangent des informations lorsqu'ils modifient leurs données. La troisième phase sert à interrompre les échanges d'information pour terminer une configuration valide avant de passer à une nouvelle configuration. Les données manipulées par les agents doivent atteindre un état cohérent. La dernière phase est celle de la restructuration de la coopération et de la mise en place de la nouvelle configuration valide (figure IV.5).

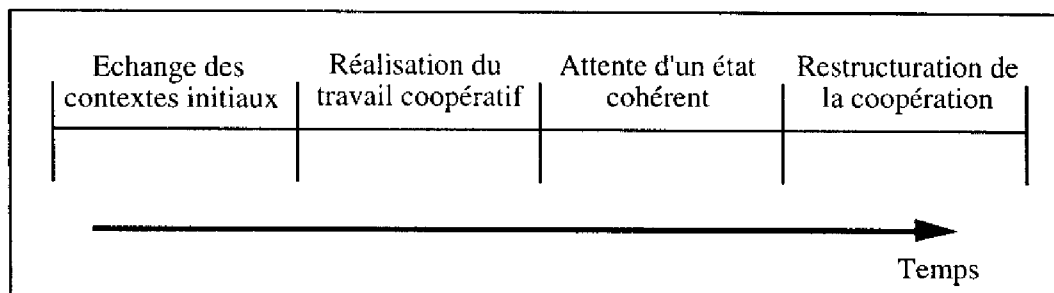


Figure IV.5. Les quatre phases pour l'évolution de la coopération

2.1.3. Implantation du protocole et du service de coopération

Le service de coopération, fourni à chaque agent utilisateur de la coopération, a été implanté au moyen d'un ensemble de modules développés en Estelle [ISO9074] puis traduits de façon semi-automatique en C [VIL95] [OWE96d]. Chaque utilisateur (le module interface de coopération) se connecte à son module protocole de coopération (figure IV.2) et communique avec lui au moyen d'un ensemble de primitives de service définies dans [DIA93d], [VIL95]. Ces derniers se chargent d'acheminer les requêtes d'entrée et de sortie de la coopération provenant des agents coopératifs. Ils transmettent les messages de vote, puis assurent la synchronisation des agents du groupe lors des changements de configuration de coopération.

Le gestionnaire central de la coopération (figure IV.2) contrôle l'ensemble de la dynamique du groupe coopératif. Il prend en compte les requêtes d'entrée et de sortie de la coopération, organise les prises de décision au moyen de votes et surveille l'enchaînement des phases de coopération lors du changement de configuration valide. En suivant les règles de la coopération situées au niveau applicatif, il garantit le bon déroulement de la dynamique du groupe. Les modules protocoles de coopération et le gestionnaire communiquent entre eux en utilisant un service de diffusion fiable réalisé au dessus du protocole TCP. Ce service doit diffuser ses messages sans perte et de manière FIFO entre un émetteur et un des récepteurs. La diffusion est réalisée à l'aide de N connexions TCP ayant toutes la même extrémité émettrice. Ainsi, grâce aux propriétés du protocole TCP, les diffusions sont fiables et ordonnées sur chacune des connexions entre l'émetteur et un récepteur [OWE96d].

2.2. L'interface de coopération

L'interface de coopération est en contact à la fois avec l'utilisateur et avec le protocole de coopération. Elle permet à un utilisateur d'effectuer une requête pour entrer en coopération ou pour quitter cette coopération grâce à une interface X Window / Openlook ou X Window / MOTIF. Cette interface informe l'utilisateur de la configuration actuelle du groupe, des relations qu'il peut avoir avec les autres participants (suivant le graphe de coopération), et lui permet de voter lors de demandes de modification du groupe. De plus, cet agent est en contact, par un *tube nommé* [OWE96d], avec la partie visioconférence de N-TSVS, ce qui permet à N-TSVS de réagir en fonction des informations de coopération qu'elle reçoit (par exemple, ouvrir de nouveaux points de communication, de nouvelles fenêtres vidéo si de nouveaux participants entrent, ou les fermer s'ils sortent) et d'informer la partie coopération des activités et de l'état de la visioconférence.

L'interface de coopération sert donc de plaque tournante entre l'utilisateur, le protocole de coopération et la visioconférence ; elle doit donc prendre en compte toutes les requêtes provenant de ces 3 entités et les diffuser vers les autres entités concernées. La machine à états décrivant le fonctionnement de l'interface de coopération [VIL95][HAM96] est représentée sur la figure IV.6. Cependant, pour ne pas surcharger la figure, seuls les événements provenant du service de coopération et de l'utilisateur, via le protocole X, sont représentés ; les messages échangés avec la partie visioconférence de N-TSVS n'apparaissent pas car N-TSVS ne fait que réagir aux sollicitations de l'interface de coopération et n'influe donc pas sur son comportement ; toutefois, dans la partie 2.4, nous montrerons sur un exemple tous les échanges entre l'interface de coopération, le service de coopération et N-TSVS.

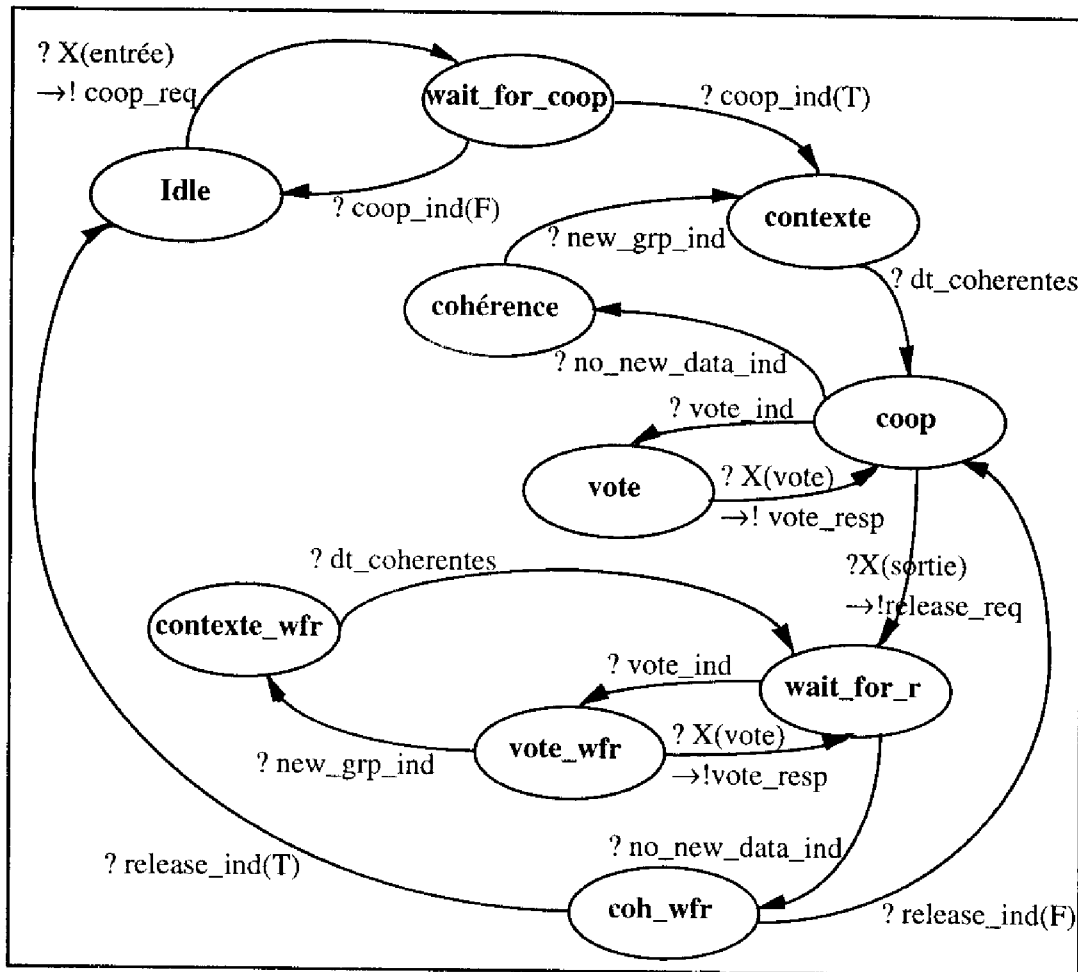


Figure IV.6. Machine à états de l'interface de coopération

Cette machine à états comporte deux états stables qui sont IDLE lorsque l'utilisateur n'est pas en coopération, et COOP lorsqu'il est en coopération. A partir de IDLE, l'interface peut passer dans l'état wait_for_coop si l'utilisateur demande à entrer en coopération. Si l'entrée est refusée, l'interface retourne à l'état IDLE, sinon elle passe dans l'état d'échange des contextes, puis dans l'état COOP. De là, suite à des demandes d'entrée ou de sortie de coopération d'autres utilisateurs, l'utilisateur local via son interface peut être amené à voter ; de même, l'interface peut avoir à stopper les communications pour se mettre en cohérence avec les autres utilisateurs, et à mettre à jour sa base de données relative aux participants au travail coopératif.

A partir de l'état COOP, et suite à une demande de sortie de coopération d'un utilisateur, l'interface passe à l'état wait_for_r (wait_for_release), et peut, grâce à un protocole quasi similaire à celui décrit pour une entrée en coopération, sortir de la coopération.

Toutefois, à cause de son rôle de plaque tournante entre 2 modules et un utilisateur, il est difficile de bien cerner le comportement de l'interface de coopération suivant le contenu et la provenance des informations qu'elle reçoit. Aussi, la partie 2.4 présente un chronogramme des informations qui transitent entre l'interface de coopération, le service de coopération et la partie visioconférence de N-TSVS pour une entrée suivie d'une sortie de coopération de l'utilisateur.

2.3. La coopération dans la partie traitement des données de N-TSVS

Pour pouvoir s'adapter aux évolutions du groupe coopératif, N-TSVS est en contact permanent avec l'interface de coopération par l'intermédiaire d'un canal FIFO. En effet, lorsque le groupe est modifié (arrivée ou sortie d'un agent), la partie visioconférence est modifiée pour prendre en compte cette adhésion au groupe ou ce départ. Ainsi, N-TSVS est pilotée par un processus (en contact avec la partie coopération) responsable de la réservation (resp. la libération) des ressources et de la création (resp. destruction) des entités nécessaires à la prise en compte du nouvel arrivant (resp. du nouveau sortant).

N-TSVS va donc évoluer en fonction des informations de coopération qui lui parviennent ; ses évolutions respectent la machine à états représentée sur la figure IV.7. De plus, elles recourent les évolutions des machines à états de la partie coopération, et en particulier de l'interface de coopération avec laquelle elle est en contact. Ce sont les primitives d'interconnexion de l'interface de coopération et de la visioconférence présentées dans [OWE96d] qui permettent à l'interface de coopération de communiquer les modifications de groupe à N-TSVS, et de contrôler ses évolutions ; la machine à état de N-TSVS en est entièrement dépendante.

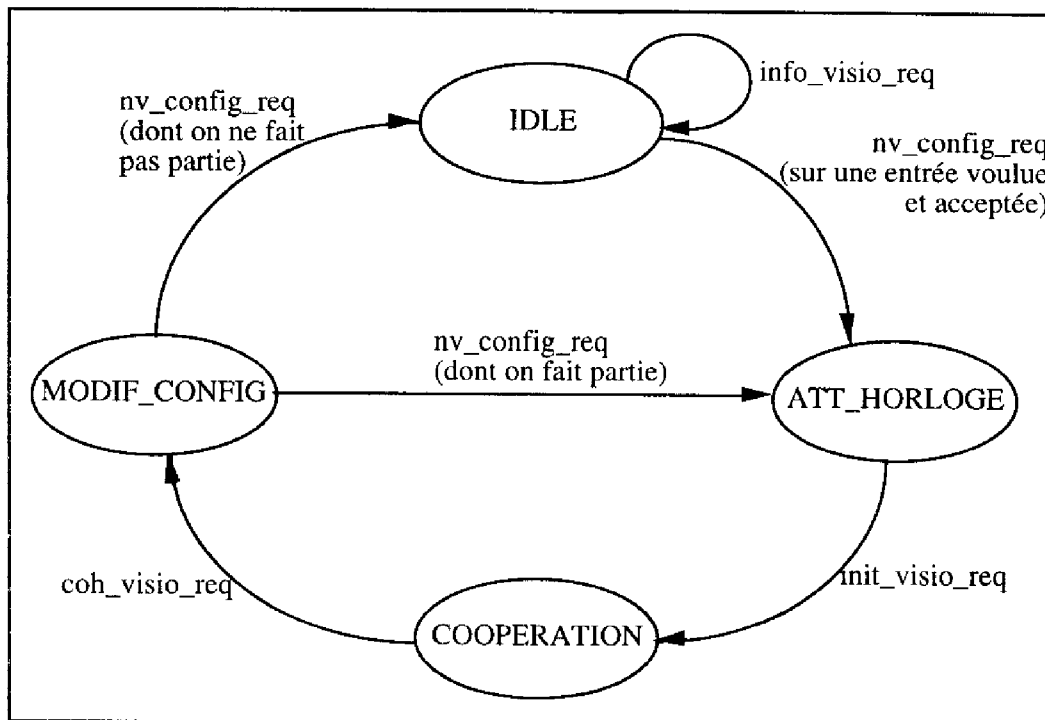


Figure IV.7. Machine à états de N-TSVS

Description de la machine à états :

A partir de l'état IDLE qui indique que cette entité de la visioconférence n'est pas en coopération, N-TSVS peut recevoir de l'interface de coopération la requête `info_visio_req`, suite à une demande d'entrée en coopération de l'utilisateur (humain) de cette application. N-TSVS répond à cette requête en précisant la qualité de service qu'il demande. Si cette demande d'entrée est refusée, N-TSVS reste dans l'état IDLE, l'utilisateur humain étant averti de ce refus par l'interface de coopération. Sinon, l'utilisateur est averti de son acceptation par l'interface de coopération qui indique de même à N-TSVS, grâce à la requête `nv_config_req`, son acceptation au groupe coopératif, lui transmet la liste des membres déjà en coopération, et la QoS effective pour le travail de groupe. N-TSVS initialise alors sa structure de données, crée ses

points de communication, acquitte la requête reçue (pour éviter de recevoir d'autres requêtes dans un état transitoire) et passe dans l'état ATT_HORLOGE.

Dans l'état ATT_HORLOGE, N-TSVS attend pour commencer les traitements audio et vidéo une requête init_visio_req qui contient l'heure globale du système. Cette horloge est nécessaire car la synchronisation multimédia (voir paragraphe 3.3) est réalisée en utilisant des estampilles temporelles. Dès que l'heure globale du système est reçue, N-TSVS crée les entités de traitement audio et vidéo, initialise son horloge locale sur l'heure globale du système, acquitte la réception de l'heure globale et passe dans l'état COOPERATION correspondant à une phase de travail coopératif stable.

Dans l'état COOPERATION, suite à une demande d'entrée ou de sortie (locale ou distante) acceptée par la partie coopération et le gestionnaire central de coopération, la requête coh_visio_req est reçue ; cette requête permettrait à N-TSVS de mettre ses données dans un état cohérent avant une modification de groupe. Toutefois, comme N-TSVS ne manipule que des données évanescences, cette requête n'est pas essentielle. N-TSVS y répond au plus tôt et passe dans l'état MODIF_CONFIG.

Dans l'état MODIF_CONFIG, N-TSVS ne peut recevoir qu'une requête nv_config_req. Toutefois, si l'entité locale ne fait pas partie de la nouvelle configuration du groupe coopératif (i.e. l'utilisateur humain local a demandé à quitter la conférence et sa requête a été acceptée), alors N-TSVS détruit toutes les entités de traitements audio et vidéo et les points de communication, re-initialise sa structure de données, et passe à l'état IDLE après avoir acquitté la réception de la requête nv_config_req. Si le participant local fait toujours partie de la coopération, alors N-TSVS reconfigure ses points de communication (en crée dans le cas d'une entrée, en détruit dans le cas d'une sortie), acquitte réception de la requête et passe dans l'état ATT_HORLOGE.

2.4. Interactions entre les modules

Comme vu plus haut, les interactions entre modules se font par messages. Les machines à états des différents modules sont complexes, et le nombre de combinaisons qui peuvent exister est bien trop important pour pouvoir représenter le comportement du logiciel dans son ensemble¹. Toutefois, dans certains cas simples, il est possible de dessiner un chronogramme décrivant l'évolution des divers modules ; sur la figure IV.8, une séquence normale d'entrée en coopération suivie d'une sortie d'un groupe de coopération déjà formé est représentée. En fait, on s'intéresse plus précisément dans ce cas aux deux parties hautes, à savoir l'interface de coopération et la partie visioconférence de N-TSVS.

Au départ de la séquence, l'interface et N-TSVS sont inactives (IDLE). L'utilisateur actionne le bouton d'entrée en coopération, ce qui lance le protocole suivant : tout d'abord, l'interface envoie une requête info_visio_req à N-TSVS afin de s'informer des paramètres de QoS de la visioconférence. Puis, elle demande à entrer en coopération au service de coopération ; elle se place dans l'état wait_for_coop et attend la réponse (positive dans le cas présent). Dès lors, elle demande l'initialisation des moyens vidéo à N-TSVS, puis attend que tous les modules aient mis leurs données dans un état cohérent. Une fois cette phase terminée, l'interface donne le signal de départ à N-TSVS (en lui transmettant notamment l'horloge globale du système qui lui a été fournie par le gestionnaire central, et qui permet, entre autre, de mettre en place les mécanismes de synchronisation multimédia), et la session démarre à la fois pour N-TSVS et l'interface qui passent en état de coopération.

La séquence pour sortir de coopération est de même nature.

¹Seules des descriptions formelles et des vérifications du système composant par composant ont été réalisées, notamment en utilisant le logiciel de validation VAL [VIL95].

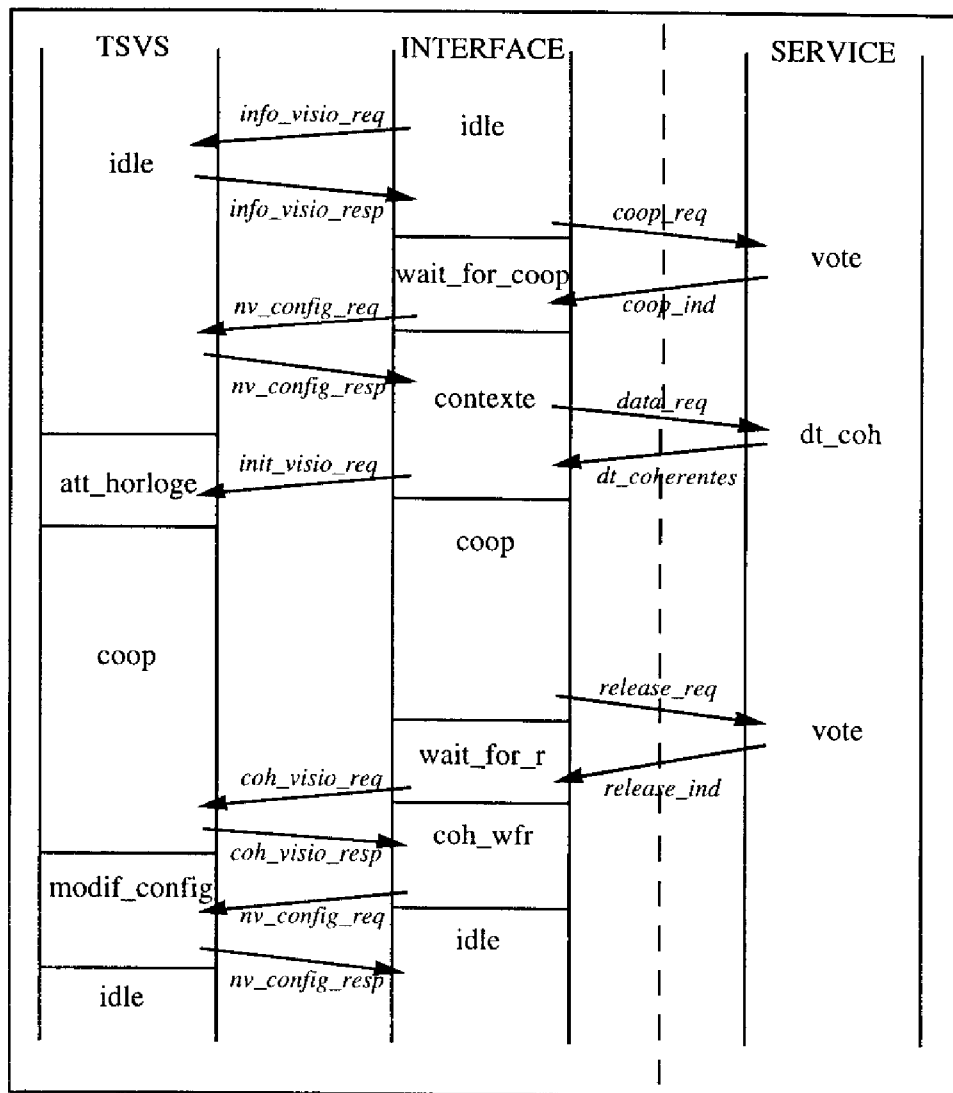


Figure IV.8. Séquence normale d'entrée et de sortie d'un groupe coopératif

3. La partie données de la visioconférence

Dans notre implémentation actuelle de N-TSVS, nous avons admis que tous les participants à la visioconférence peuvent communiquer avec tous les autres, qu'il n'existe pas de hiérarchie explicite entre ces participants, et que toutes les configurations des participants sont valides : en fait, cette architecture de coopération correspond à une activité totale (figure IV.1)².

Cette partie "donnée" propre à la visioconférence (la partie coopération pouvant apparaître dans plusieurs types d'applications coopératives) s'occupe des tâches suivantes :

- les communications multipoints,
- les traitements des données audio et vidéo,
- la synchronisation multimédia,
- la gestion et la mise en oeuvre des mécanismes permettant de respecter la QoS à N.

²L'intégration de mécanismes permettant la prise en compte de groupes structurés au niveau de la partie "données" de N-TSVS sera vue dans la partie 5 de ce chapitre.

3.1. Les communications multipoints

Pour pouvoir communiquer de façon efficace (i.e. en consommant un minimum de ressources machine et réseau) avec plusieurs partenaires simultanément, un service de diffusion (ou service multicast) est requis. Aussi, deux bibliothèques de communications multipoints ont été écrites : une bibliothèque de diffusion fiable (`reliable_mcast`) pour les informations de signalisation, et une bibliothèque de diffusion non fiable (`unreliable_mcast`) pour les données de la visioconférence. Ces deux bibliothèques utilisent le service réseau IP multicast [DEE88] qui reste le seul protocole de diffusion à peu près universellement distribué.

La bibliothèque de diffusion `unreliable_mcast` reprend les fonctionnalités offertes par IP étendu pour le multicast, mais propose une interface d'utilisation simplifiée ; par exemple, cette bibliothèque se charge de la formation des adresses de classe D pour les groupes de diffusion, ce qui allège le travail de l'utilisateur d'une tâche parfois délicate.

La bibliothèque de diffusion fiable `reliable_mcast` fiabilise les communications en utilisant un algorithme basé sur l'acquittement positif et la réémission sélective ; en effet, chaque destinataire d'un message acquitte au plus tôt le message qu'il vient de recevoir (ce qui n'est pas pénalisant car la bibliothèque de diffusion fiable n'est utilisée que pour des messages de signalisation peu fréquents dans N-TSVS). De plus, comme les supports de communication actuels sont relativement fiables, le nombre de pertes sera limité, et il est judicieux de ne pas réémettre les données par un mécanisme de diffusion (qui surchargerait inutilement tous les destinataires), mais plutôt par des communications point à point. Toutefois, pour pouvoir gérer ce mécanisme, l'émetteur d'un message doit disposer de la liste des adresses unicast de chaque destinataire, la gestion des acquittements et des réémissions se faisant à partir de cette liste. Il est à noter qu'avec ce mécanisme, si la liste des destinataires ne comporte pas tous les membres du groupe de diffusion, les communications vers la liste des destinataires précisée en paramètre seront fiables alors que pour les autres membres du groupe de diffusion elles seront non fiables ; ceci permet donc de réaliser un système de diffusion qui s'apparente à un service de diffusion à la fois fiable et non fiable selon 2 sous-ensembles d'utilisateurs [REZ96].

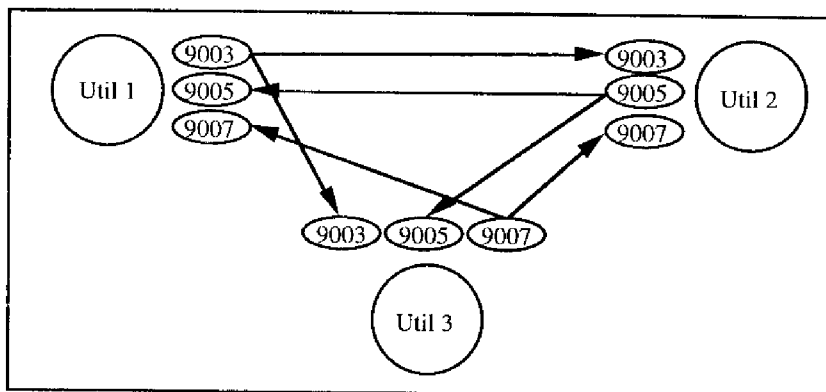


Figure IV.9. Architecture des points de communication dans N-TSVS

En utilisant ces bibliothèques, et parcequ'il a été choisi que certains participants pouvaient ne participer à une séance de visioconférence que par l'intermédiaire du média audio, deux groupes de diffusion sont utilisés : un groupe de diffusion des données audio, et un groupe de diffusion des données vidéo. Au sein de chacun de ces groupes, l'architecture de sockets représentée sur la figure IV.9 est utilisée (la figure IV.9 représente plus particulièrement l'architecture des points de communication du groupe vidéo). Dans cette architecture, chaque utilisateur ouvre N points de communication (N est le nombre de participants à la conférence) : tous ces points possèdent un numéro de port particulier qui est fonction de l'identité de l'utilisateur avec lequel il sont en relation. Par exemple, sur la figure IV.9, l'utilisateur 1 émet ses

données vidéo sur le port 9003, et chacun des autres adhérents à ce groupe ouvre un point de communication sur ce port pour recevoir les données vidéo émises par l'utilisateur 1. De même, l'utilisateur 2 émet ses données vidéo sur le port 9005, et les utilisateurs 1 et 3 ouvrent donc des points de communication sur ce port pour recevoir les données vidéo émises par 2. La règle utilisée pour attribuer les numéros de port consiste à affecter pour les communications de l'utilisateur i ou avec l'utilisateur i le port $9000 + 2i + 1$ (les ports pairs $9000 + 2i$ sont attribués pour les points de communication audio).

3.2. La synchronisation multimédia dans N-TSVS

Comme dans TSVS, la synchronisation dans N-TSVS est obtenue grâce à une méthode basée sur l'utilisation d'estampilles. Cette méthode consiste à attribuer à chaque objet capturé sur la machine émettrice une estampille, et à présenter cet objet sur les machines réceptrices à la date indiquée par cette estampille. Cette méthode de synchronisation très simple respecte en fait le paradigme de la ligne temporelle et assure donc à la fois les synchronisations intra et inter-flux (cette méthode résout notamment les problèmes de pertes, de déséquences et de duplications des objets par le système de communication).

Pour que la technique de synchronisation par estampilles fonctionne correctement, il est nécessaire de synchroniser les horloges de toutes les machines (aux phénomènes d'asynchronisme du système de communication près). En fait, dans la méthode utilisée, le gestionnaire de coopération est le maître du temps global, et régulièrement (lors de chaque changement de coopération), il demande à tous les participants de la conférence de synchroniser leurs horloges avec la sienne, en leur transmettant son horloge locale.

De plus, à cause de la finesse des contraintes temporelles sur les flux de la visioconférence, le module de traitement des données de N-TSVS s'appuie sur une architecture "multi-threads" qui permet seule de garantir cette finesse de synchronisation des flux multimédias en offrant un important degré de parallélisme. L'architecture des threads de N-TSVS est représentée sur la figure IV.10 ; sur cette figure, seuls les threads de traitement des données vidéo sont représentés (pour accroître la lisibilité du schéma). Toutefois, il existe autant de threads audio que de threads vidéo, et autant de points de communication audio que de points de communication vidéo.

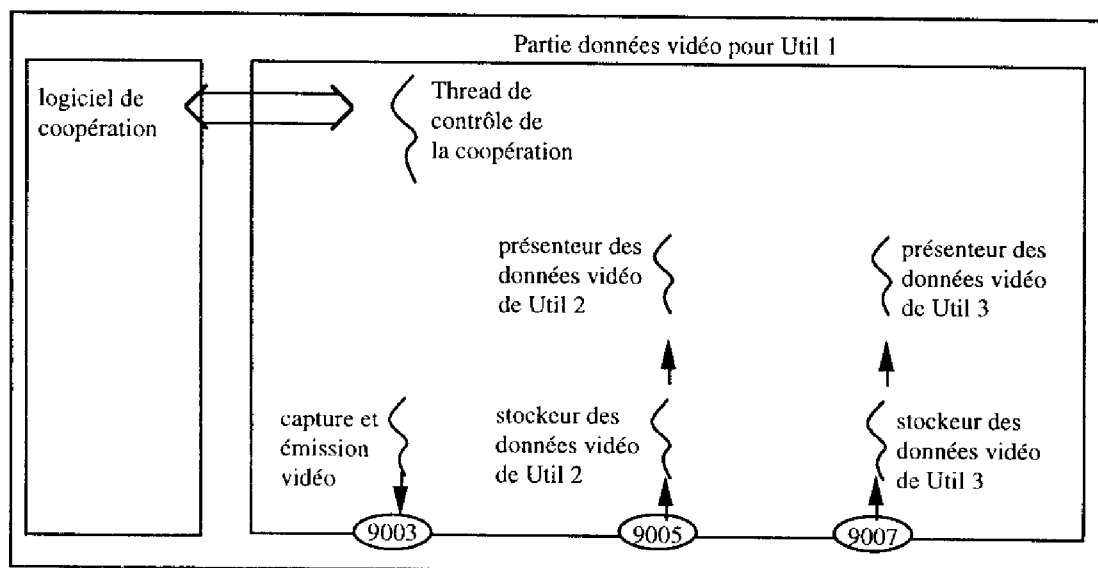


Figure IV.10. Architecture des threads vidéo de N-TSVS

Sur cette figure apparaissent un thread de capture / émission pour les données vidéo de la partie locale de la visioconférence, et pour chaque flux de chaque participant un thread de réception et un thread de présentation des données provenant du participant en question. Le thread de capture / émission vidéo se réveille régulièrement (la fréquence dépendant de la qualité vidéo demandée par l'utilisateur), capture l'image du moment, la compresse au format JPEG, l'estampille en fonction du dernier temps global reçu du gestionnaire central et l'émet sur le réseau. Le thread de réception, par contre, est en écoute permanente sur le point de communication auquel il est affecté, et stocke chaque image reçue dans un tampon. Le thread de présentation, quant à lui, scrute le tampon auquel il est associé, et lorsqu'une image est stockée, il la décompresse et la présente à la date indiquée par l'estampille plus un temps constant de temporisation. Le fonctionnement des threads audio est totalement équivalent à celui des threads vidéo.

Toutefois, N-TSVS, contrairement à PNSVS [OWE95c], n'offre aucune garantie quant au respect strict des contraintes de synchronisation temporelles ; le logiciel peut être qualifié de "best effort", puisqu'il implémente un algorithme de synchronisation multimédia sans se soucier du comportement asynchrone du système opératoire. Pour corriger ce défaut, la partie 3.4 montre comment l'asynchronisme des systèmes distribués peut être pris en compte pour garantir les contraintes de synchronisation multimédia.

3.3. Le multiplexage audio

Un autre problème technique important à résoudre pour réaliser N-TSVS consiste à multiplexer les $N-1$ flux audio (à 64kbps chacun) des $N-1$ interlocuteurs sur la seule sortie audio (à 64 kbps) de la station. La solution la plus proche de la réalité consisterait à faire la "somme sonore" de tous les signaux audio qui arrivent à la station pour obtenir un flux unique reproduisant l'ambiance réelle de la salle virtuelle de visioconférence. Actuellement, nous nous sommes contentés de réaliser ce multiplexage avec des lois de codages linéaires [OWE96d], et pas avec la loi de codage μ qui est souvent la seule loi connue par nos cartes audio [G711]. Pour éviter de systématiquement convertir les signaux codés en loi μ en loi linéaire, puis de nouveau en loi μ , après avoir fait le multiplexage, nous nous sommes orientés vers une autre solution. Ainsi, un filtrage à l'émission est réalisé, en effectuant une détection des blancs [OWE96d] ; seuls les objets qui ne correspondent pas à des silences sont émis. De ce fait, grâce à la synchronisation temporelle qui existe entre tous les participants, et si des utilisateurs ne parlent pas en même temps, au plus un flux audio parviendra à tous les participants à un instant donné. Si des utilisateurs parlent en même temps, le son produit est incompréhensible et une autorégulation des temps de parole des intervenants se fera.

3.4. La gestion de la QoS et de la synchronisation à N

Dans cette partie deux points vont être abordés : tout d'abord, une généralisation de la technique de gestion de la QoS au sein de groupes de travail coopératif sera proposée, puis, on montrera comment les principes de synchronisation point à point de PNSVS [OWE95a][OWE95c] peuvent être étendus pour la synchronisation multimédia et le respect de certains paramètres de QoS dans une visioconférence multipoints.

3.4.1. Gestion de la QoS à N

La partie 3.1 a montré comment la qualité de service était gérée dans N-TSVS. En fait, de par la nature point à multipoints des communications applicatives, il apparaît que la qualité de service doit être gérée sur chaque multi-connexion point à

multipoints ; ainsi, chaque multi-connexion peut avoir sa propre QoS et elle n'est pas obligée de se restreindre à une QoS globale au groupe de travail. Ainsi, pour généraliser ce qui a été fait dans N-TSVS, une QoS par multi-connexion applicative est considérée, ce qui engendre au niveau des communications IP (conformément à ce qui a été fait dans la partie 3.1) un couple (groupe de diffusion, numéro de port) par multi-connexion applicative. Ainsi, la gestion de la QoS dans un groupe coopératif étend les mécanismes de gestion de la QoS d'une connexion point à point (décrits dans [OWE95c]) pour une multi-connexion 1->N.

Cette gestion de la QoS multi-connexion par multi-connexion n'est pas vraiment adaptée au service fourni par IP étendu pour le multicast du fait de la nature datagramme de ses communications et de la notion de diffusion non structurée dans un groupe ; la gestion des multi-connexions se fait par un artifice sur les numéros de port. Toutefois, cette gestion s'adapte parfaitement aux services de diffusion en mode connecté, comme celui fourni par un réseau ATM, qui implémente physiquement des multi-connexions 1->N, respectant une QoS donnée en terme de débit, délai, etc.

3.4.2. Gestion de la synchronisation multimédia à N

La technique de synchronisation multimédia par estampilles, utilisée dans N-TSVS, ne permet pas de garantir les contraintes de synchronisation des objets audio et vidéo (cf. 3.2.). En effet, comme vu au chapitre 1, les processus de présentation (qui s'exécutent en classe temps partagé ou en classe système lors des appels système) ont un comportement asynchrone (non déterministe). De plus, comme la précision temporelle des processus temps réel du système Unix Solaris 2 n'est pas inférieure à 10 ms, la technique de synchronisation basée sur des estampilles temporelles, qui imposent des date fixes, ne peut pas s'adapter aux contraintes du système opératoire. Aussi, de la même façon que PNSVS apporte une solution aux problèmes que TSVS ne peut résoudre [OWE95c], les mécanismes de synchronisation déduits d'une modélisation TSPN permettent de garantir les contraintes de synchronisation et de QoS à N que N-TSVS et sa technique de synchronisation à base d'estampilles ne peuvent garantir. En effet, dans PNSVS, tout le travail de synchronisation multimédia et de gestion de la QoS est fait au niveau du récepteur. Chaque récepteur d'une multi-connexion 1->N peut donc utiliser ces mêmes mécanismes ; un TSPN de présentation est associé à chaque multi-connexion applicative, qui, à un niveau inférieur, se divise en un TSPN applicatif pour l'émetteur et un TSPN applicatif pour les récepteurs (comme dans PNSVS 1).

Les mécanismes de synchronisation avec 3 threads par flux et l'utilisation de la classe d'ordonnancement temps réel (un thread temps partagé de stockage, un thread temps partagé de présentation et un thread temps réel d'orchestration) permettent donc de garantir les contraintes de synchronisation des flux audio et vidéo, modélisées par les TSPN de présentation et applicatifs.

La gestion du délai de présentation de bout en bout en fonction du niveau de remplissage des tampons s'adapte également à une multi-connexion 1->N, car cette gestion ne nécessite ni dialogue entre l'émetteur et les récepteurs, ni réémission de certains objets audio ou vidéo.

L'intégration des mécanismes de gestion de la synchronisation multimédia et du délai de présentation de bout en bout ne présente donc aucune complexité conceptuelle ; ces mécanismes ne font intervenir que les récepteurs, et multiplier les récepteurs ne les modifie en rien.

Enfin, un transport multicast à ordre partiel qui offrirait le même service que celui point à point (perte et délivrance des objets au plus tôt) pourrait également améliorer les performances et la qualité de présentation de l'application de visioconférence à N - N-PNSVS - synchronisée en implémentant les mécanismes déduits des TSPN ; néanmoins, l'implémentation de N-PNSVS, qui reste à réaliser, offre une des perspectives intéressantes comme suite des travaux présentés dans cette partie.

4. Traitement des erreurs dans N-TSVS

Dans ce qui précède, N-TSVS apparaît comme un assemblage complexe de processus. Lors des tests, ces processus se sont montrés fragiles lors d'une utilisation incorrecte ou d'un manque de ressources de la machine. Par ailleurs, l'ensemble de l'édifice possède un graphe d'états très complexe et qui peut présenter des cas de dysfonctionnement ; par exemple, si un coopérant quitte brutalement l'application (en tuant les processus locaux), les autres coopérants et le gestionnaire de coopération ne sont pas avertis, et lors d'un vote (par exemple), la voix du coopérant manquant sera attendue sans fin. Toute évolution du groupe est donc rendue impossible.

Aussi, un système de détection, de propagation et de traitement des erreurs a été mis en place. Ce système ne garantit toutefois pas que toutes les erreurs seront prises en compte. En effet, l'objectif n'est pas de traiter le cas général, mais seulement de rendre le fonctionnement de cette application plus robuste pour les cas de panne les plus fréquemment observés lors des tests du logiciel. Il suffit pour cela de faire en sorte qu'une erreur sur un site ne puisse pas bloquer toute l'application, et que les autres sites puissent fonctionner dans une nouvelle configuration imposée (principe de la "panique organisée" [MEY88]).

4.1. Les erreurs dans N-TSVS

Les erreurs considérées dans N-TSVS peuvent être réparties en 3 catégories :

- le non respect du manuel d'utilisation du logiciel :
 - si l'utilisateur termine brutalement son application sans être sorti de la coopération,
 - si l'utilisateur tue un des processus de l'application,
- les erreurs (bogues) de N-TSVS :
 - mauvaise initialisation de la caméra,
 - saturation de la mémoire,
 - toute autre mort prématurée de N-TSVS,
- la mort prématurée du gestionnaire central de coopération ou des services de diffusion. Toutefois, ces pannes ont été rendues aussi peu probables que possible lors du développement de ces modules, et effectivement lors des tests, ces modules n'ont jamais été pris en faute. Aussi, il n'a pas été conçu de mécanisme de recouvrement sur ces modules.

Ainsi, les deux modules sensibles de l'application sont l'interface de coopération et la visioconférence à cause de la fragilité d'X-WINDOW et des composants vidéo (cartes, bibliothèques, serveur et protocole X,...). L'objectif de ce qui suit est donc de pouvoir détecter les mauvais fonctionnements de ces processus, et de mettre en oeuvre des mécanismes permettant à l'application de perdurer en cas d'erreur.

4.2. Le traitement des erreurs

Dans N-TSVS, les erreurs sont traitées par le service de coopération et par le gestionnaire central de coopération [VIL95] ; lorsqu'une erreur est signalée au gestionnaire central, ce dernier constitue avec les membres opérationnels une nouvelle configuration valide (des participants au travail coopératif peuvent de fait être évincés) ; il envoie alors cette nouvelle configuration valide (via un `new_grp_ind`) à tous les sites qui étaient en coopération afin qu'ils se reconfigurent en fonction de la nouvelle configuration de travail. Les erreurs pouvant se produire à tout moment, les machines à états de l'interface de coopération et de la visioconférence doivent prévoir de les prendre en compte à tout instant.

En ce qui concerne la machine à états de l'interface coopérative (cf. figure IV.6), la réception d'un message `new_grp_ind` sur un cas d'erreur ramène à l'un des états stables COOP, `wait_for_r` ou IDLE. En effet, si le participant local fait toujours partie de la nouvelle configuration et si l'interface se trouvait dans un des états `vote`, `vote_wfr`, `contexte`, `coherence` ou COOP, la reconfiguration est effectuée, et l'interface passe dans l'état COOP ; si l'interface se trouvait dans l'état `contexte_wfr`, `wait_for_r` ou `coherence_wfr`, alors après reconfiguration, l'interface passe dans l'état `wait_for_r`. Dans tous les cas, si le participant local ne fait pas partie de la nouvelle configuration, l'interface passe à l'état IDLE.

Pour la visioconférence, la nouvelle machine à états est donnée sur la figure IV.11. De la même manière que le service de coopération transmet un message `new_grp_ind` à l'interface de coopération, cette dernière transmet à la visioconférence un message `nv_config_req` précisant que la reconfiguration est due à une erreur, cette reconfiguration pouvant arriver à tout moment.

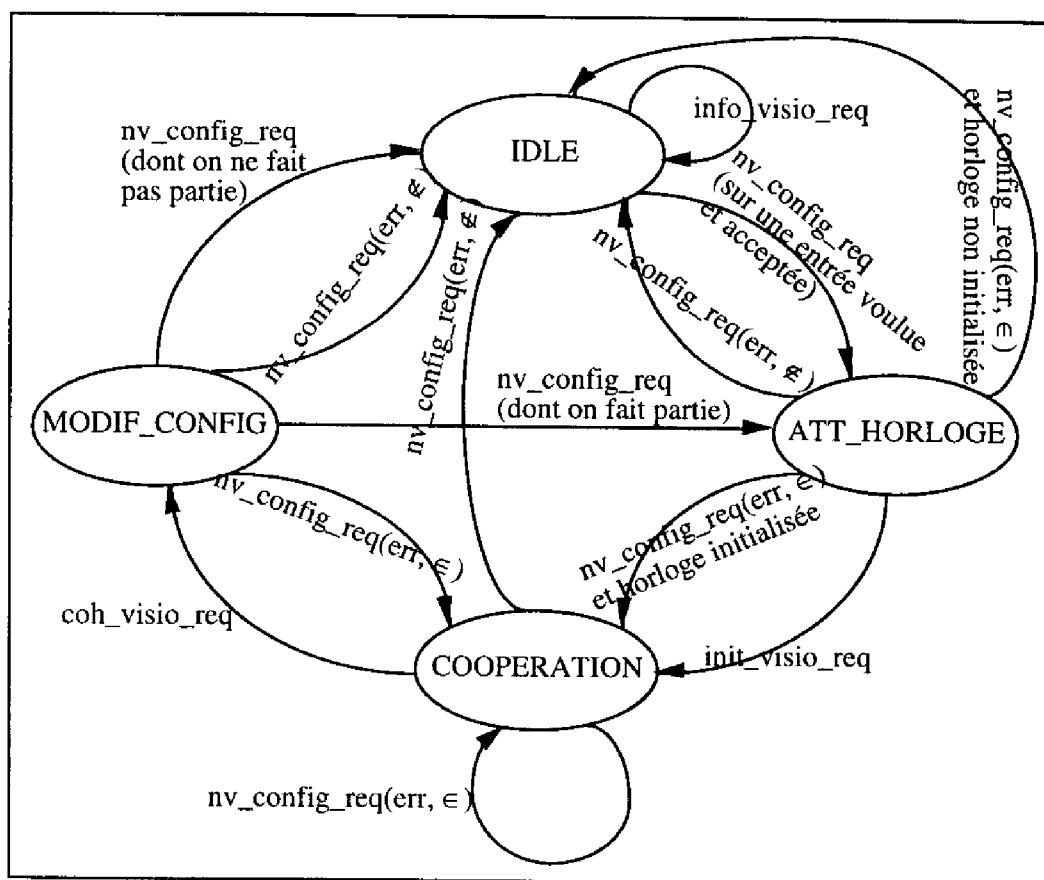


Figure IV.11. Machine à états de la visioconférence intégrant le traitement des erreurs

En fait, les cas d'erreurs qui peuvent se produire sont les suivants :

- dans les états ATT_HORLOGE, COOP ou MODIF_CONFIG, si la partie locale ne fait pas partie de la nouvelle configuration, tous les processus et points de communication sont détruits, les ressources sont libérées et la visioconférence passe à l'état IDLE.
- si le participant local fait toujours partie de la nouvelle configuration, alors depuis les états MODIF_CONFIG et COOP, la visioconférence se reconfigure et passe à l'état COOP.
- Par contre, dans l'état ATT_HORLOGE, deux traitements différents apparaissent ; en effet, il est important pour les mécanismes de synchronisation que l'horloge locale ait

été initialisée. Aussi, si le participant local était déjà en coopération, son horloge est initialisée, et il peut continuer à travailler avec les participants de la nouvelle configuration. Par contre, si la visioconférence était précédemment dans l'état IDLE, si la requête d'entrée en coopération a été traitée par le gestionnaire central et si le site local fait partie de la nouvelle configuration, l'absence d'horloge ne permettra pas à la visioconférence de traiter de façon correcte les données de l'application. Aussi, dans ce cas, un message de déconnexion brutale est généré, et la visioconférence retourne à l'état IDLE³.

4.3. Détection et propagation des erreurs locales

Une fois le problème du traitement des erreurs résolu grâce à une méthode basée sur une reconfiguration ordonnée par le gestionnaire central, il faut pouvoir, a priori, signaler à ce même gestionnaire les erreurs qui se produisent sur les différents sites. Comme vu précédemment, les deux modules sensibles sont l'interface de coopération et la visioconférence, et il faut donc, lors de la mort non souhaitée d'un de ces deux modules, transmettre un message d'erreur au gestionnaire de coopération. Dans les premières tentatives, vite abandonnées, nous avons essayé d'utiliser la primitive `at_exit` qui permet d'appeler une fonction lorsque le processus se termine ; ainsi, nous pensions pouvoir émettre un message d'erreur vers le gestionnaire central à la fin de nos processus sensibles. Toutefois, ceci s'est avéré inefficace, car le `at_exit` ne fonctionne que sur une terminaison programmée des processus.

Une autre solution a donc dû être envisagée. Le problème majeur dans la détection de la mort des processus vient de leur caractère imprévisible. La solution la plus simple qui nous est apparue consiste donc à détecter la mort des processus au niveau d'un processus père. C'est pourquoi, les processus de l'interface de coopération et de la visioconférence sont lancés par un processus père qui n'a ensuite pour rôle que d'attendre leur terminaison (par la primitive `wait`). Aussi, en cas de mort d'un processus suite à une erreur, c'est le père qui se charge d'envoyer le message d'erreur vers le gestionnaire central de coopération.

Ce message ultime n'est cependant pas garanti ; si la mort d'un processus est due à un problème de communication entre le site local et le gestionnaire central, le message a peu de chances de parvenir à son destinataire. De même, en cas d'arrêt de la machine, tous les processus de cette machine meurent, y compris le père de l'interface de coopération et de la visioconférence ; le message d'erreur ne sera donc pas émis vers le gestionnaire central ; toutefois, ce dernier cas est hors propos, car ce genre d'erreur n'était pas pris en compte dans les hypothèses, et nécessiterait pour être résolu des mécanismes d'un tout autre niveau de complexité. Le procédé décrit et implémenté se révèle néanmoins utile en cas de mort voulue ou non d'un des processus sensibles, et le mécanisme de reconfiguration sur erreur permet à l'application de continuer à fonctionner.

5. Structuration de la visioconférence

Le dernier point traité dans ce chapitre consiste à étendre la visioconférence en une application structurée. En effet, la partie coopération considère que l'activité coopérative est une activité structurée avec des échanges qui doivent être contrôlés. Pour le moment, N-TSVS a été conçue comme une application pour des activités dont

³Plutôt que de traiter ces deux cas en fonction de la valeur d'une variable (comme cela a été fait ici), il aurait été possible de diviser l'état `ATT_HORLOGE` en deux - `ATT_HORLOGE1` qui ne peut être atteint que depuis `IDLE` et `ATT_HORLOGE2` qui ne peut être atteint que depuis `MODIF_CONFIG` - de sorte que le traitement en cas d'erreur dans `ATT_HORLOGE1` soit une déconnexion brutale et dans `ATT_HORLOGE2` un passage à l'état `COOP` après reconfiguration.

le graphe de coopération est totalement maillé, et elle ne peut pas s'adapter aux cas où le graphe coopératif n'est pas total.

En fait, le problème vient de la nature même du service de diffusion d'IP multicast qui permet de communiquer avec tous les membres d'un groupe coopératif sans restriction (l'adresse du destinataire est une adresse de groupe qui ne prend pas en compte des adresses individuelles).

La solution directe consiste donc à écrire une librairie au dessus d'IP multicast offrant les fonctionnalités manquantes en terme de service⁴. En fait, il suffit d'utiliser le même principe que pour la librairie de diffusion fiable vue dans la partie 3.1, à savoir utiliser une liste des destinataires du message au sein du groupe⁵. Pour cela, il faudra transmettre cette liste dans le message (contrairement à ce qui est fait dans la librairie `reliable_mcast` où elle ne sert qu'à analyser les acquittements retournés), et réaliser un filtrage au niveau de la primitive de réception, de sorte que le message ne soit délivré à l'utilisateur que si ce dernier est dans la liste de diffusion. Ainsi, on obtient à faible coût un service de diffusion multicast structuré.

Conclusion

Ce chapitre a présenté une approche générique dont le but est de concevoir et implémenter des applications multimédias distribuées coopératives. Pour cela, un service général de gestion de la coopération commun à toutes les applications coopératives a été isolé. Ensuite, une application de visioconférence, utilisant ce service de coopération a été présentée. Cette application a été implémentée sur une plate-forme à base de stations de travail Sun Solaris 2 équipées de cartes vidéo Parallax, et interconnectées par des réseaux de type Ethernet et ATM. La partie gestion de la coopération a été écrite en Estelle, puis traduite en C [VIL95] [OWE96d]. La partie traitant les données de la visioconférence a été, quant à elle, directement écrite en C, de même que les protocoles et services de diffusion.

La partie contrôle de l'ensemble de la coopération repose actuellement sur un gestionnaire centralisé. Plusieurs travaux ont cherché à améliorer et à distribuer ce contrôle en le considérant comme un jeton circulant parmi les participants. Les algorithmes permettant de distribuer le gestionnaire ont été conçus et validés [VER95] ; cependant, vu la complexité de la solution centralisée, et les difficultés supplémentaires induites par la distribution, l'implémentation d'une solution distribuée dans le cadre d'une application réelle comme N-TSVS conduit à un logiciel de très grande complexité, et dépasse largement le cadre de cette thèse.

⁴Cette fonctionnalité aurait également pu être prise en charge par la visioconférence ; toutefois il semble judicieux de traiter cette fonctionnalité au niveau le plus bas, et alléger ainsi la partie visioconférence déjà très complexe.

⁵Cette liste des destinataires est connue de la partie visioconférence de N-TSVS car elle est transmise par l'intermédiaire de la primitive `nv_config_req`, dans le champs indiquant la liste des suivants dans le graphe coopératif [OWE96d].

Chapitre V

Vers la conception et le développement d'un système synchrone de téléformation professionnelle

Introduction

L'essor des applications distribuées multimédias coopératives, et en particulier celui des applications de visioconférence, tend à ouvrir les portes du télétravail et du travail coopératif dans un grand nombre de domaines. L'un des domaines les plus actifs depuis l'explosion des télécommunications est certainement le domaine du télé-enseignement et de la téléformation. Déjà, bien avant que l'on parle de réseaux de communication numériques à hauts débits, le besoin de téléformation s'était fait sentir, et les cours par correspondance étaient apparus. Aujourd'hui, avec les nouveaux supports de communication et les nouveaux services qu'ils fournissent (courrier électronique, visioconférence, partage d'applications, etc.) les possibilités semblent sans limite, et beaucoup d'expériences ont été réalisées dans ce domaine. Ces expériences avaient pour but d'étudier le meilleur moyen d'utiliser ces nouvelles possibilités et éventuellement de développer les outils nécessaires pour obtenir des systèmes de télé-enseignement ou de téléformation les plus interactifs et efficaces possibles.

Rapidement, la société Airbus Training s'est intéressée aux systèmes de téléformation pour la formation des pilotes et agents de maintenance sur la gamme des avions Airbus. Les premiers travaux ont été menés dans le cadre du projet CESAME, et se poursuivent dans le cadre du projet TOPASE. L'objectif est de proposer, dans un environnement distribué, des outils permettant de recréer les interactions qui existent entre un instructeur et les stagiaires dans une salle de cours. Cependant, Airbus assurait déjà la formation des personnels des compagnies clientes par le biais du système d'EAO multimédia VACBI¹, mais les déplacements des stagiaires et des instructeurs vers les centres de formation et l'impossibilité d'assurer un suivi de formation efficace ont introduit l'intérêt de la téléformation. Le but de CESAME a donc été de définir un environnement de téléformation synchrone dans lequel, bien que l'instructeur et les

¹VACBI : Video and Audio Computer Based Instruction

stagiaires ne soient pas géographiquement en un même lieu, ils puissent avoir des échanges directs, à la manière d'un dialogue entre les différents intervenants. Le support privilégié de ce type de téléformation est alors le réseau de communication. Toutefois, nous nous sommes intéressés essentiellement aux outils permettant la mise en oeuvre de télé-formations synchrones, la définition des cours eux-mêmes (contenu et présentation) ne faisant pas l'objet du travail de cette thèse.

Ce chapitre est structuré de la façon suivante :

- la première partie présente une étude des systèmes de téléformation qui existent aujourd'hui ;
- la seconde partie présente le cadre de l'étude - la téléformation dans le domaine de l'aéronautique - , l'application d'EAO actuelle (VACBI) ainsi que les extensions que nous projetons d'y ajouter ;
- à partir des interactions observées entre un instructeur et ses stagiaires dans un contexte de cours VACBI, la partie 3 présente les fonctionnalités de base à offrir dans un contexte distribué ;
- ces fonctionnalités sont ensuite utilisées pour construire les outils nécessaires tant à l'instructeur qu'aux stagiaires (partie 4) ;
- la partie 5 présente alors l'intégration des outils dans un environnement commun, puis les interfaces de cet environnement pour l'instructeur et les stagiaires ;
- dès lors, la partie 6 présente les implémentations réalisées, et en conclusion nous présenterons les perspectives d'enrichissement de cet environnement et d'utilisation de celui-ci.

1. Présentation de quelques exemples de systèmes d'EAO et de télé-enseignement

Ces dernières années, le domaine de l'enseignement assisté par ordinateur (EAO) a connu une importante explosion, due aux nouvelles possibilités offertes par les ordinateurs. De très nombreux projets ont débuté, et les possibilités offertes par les technologies disponibles ont été testées ; cette attitude tend à perdurer aujourd'hui avec l'exploration des solutions offertes par les technologies multimédias et les futures autoroutes de l'information. Cette partie va donc faire une présentation sommaire de quelques systèmes d'EAO et de télé-enseignement existants ou à l'état de prototype, et une classification de ces systèmes sera proposée.

1.1. Les systèmes d'enseignement assisté par ordinateur

1.1.1. Les systèmes d'enseignement assisté par ordinateur "textuels"

La première classe des systèmes d'EAO se compose de cours textuels implémentés sur des ordinateurs, et que les apprenants peuvent consulter de façon contrainte ou non, comme c'est le cas par exemple pour le manuel UNIX. Cependant, très rapidement se pose le problème de l'assistance de l'apprenant en phase d'apprentissage avec des systèmes textuels aussi simples. Par exemple, le système CO-EXPLAINER [HOS93] a été conçu pour présenter des cours à des étudiants en respectant un ordre de parcours logique, mais également pour poser des questions aux étudiants, répondre à leurs questions ou corriger des exercices. Ceci est rendu possible par l'utilisation de techniques tirées de l'intelligence artificielle. Toutefois, il est très difficile de transformer les machines réactives actuelles en systèmes interactifs capables de remplacer les enseignants humains, et le nombre de questions auxquelles CO-EXPLAINER sait répondre est très limité.

1.1.2. Les systèmes d'enseignement assisté par ordinateur "multimédia"

De plus, les systèmes textuels souffrent cruellement d'un manque de représentation des informations qu'ils doivent faire connaître et sont très rapidement ennuyeux pour les apprenants qui doivent alors faire de gros efforts de concentration pour des résultats parfois peu probants. Aussi, pour des présentations plus agréables, et grâce aux nouvelles possibilités offertes par les technologies multimédias, de plus en plus de nouveaux systèmes d'EAO intègrent des présentations multimédias avec textes, graphiques, audio et vidéo. L'apparition du multimédia est une évolution majeure dans les systèmes d'EAO qui les rend beaucoup plus conviviaux et interactifs, notamment en utilisant des mécanismes hypermédias. Parmi les très nombreux systèmes de cette classe, il est possible de citer SteelDEM, [AMI93], TRAIN [AUG93], CALSA [BAR93a] ou VACBI [OWE95b]. Ces 4 systèmes ont une structure hypermédia avec des noeuds composés par des documents multimédias et reliés entre eux par des liens. En fait, souvent ces systèmes implémentent une métaphore de livre avec des chapitres, paragraphes, ... ou des sections, sous-sections, ...

1.2. Les systèmes de télé-enseignement

Cependant, malgré l'effort de convivialité apporté par ces systèmes d'EAO multimédias, ils restent toujours faibles au niveau de l'assistance de l'apprenant pendant la consultation du cours. En effet, l'apprenant peut se retrouver bloqué sans aucune solution lui permettant de progresser, et le mieux, dans ce cas là, est bien sûr d'avoir recours à un enseignant soit local comme dans VACBI [OWE95b], mais surtout - et c'est là que se situe l'intérêt réel de ce genre de systèmes - distant. Les prémices de ce type de systèmes sont apparus sous la forme de cours par correspondance, les Postes servant alors de médium de communication pour échanger des supports papiers ou, depuis l'explosion des technologies multimédias, des disquettes, cassettes VHS et/ou CD-ROM [GRE94] [BEL94]. Toutefois, avec l'essor des moyens de communication, et pour améliorer les temps de réponse liés à la lenteur des communications postales, des supports de communications comme le téléphone et le Fax sont aujourd'hui couramment utilisés [ELD94].

Toutefois, toutes les approches présentées ci-dessus sont lentes, lourdes et ne sont absolument pas intégrées. C'est pourquoi, l'évolution actuelle des réseaux de communication informatiques (autoroutes de l'information) apporte au domaine du télé-enseignement les possibilités qui faisaient défaut jusqu'alors. Ces autoroutes et leurs applications (e-mail, visioconférence, W3...) vont permettre d'avoir réellement des interactions entre des apprenants et des enseignants, et le terme télé-enseignement va enfin pouvoir prendre un sens concret en permettant vraiment des cours à distance [BAT94] [ELD94].

1.2.1. Systèmes de télé-enseignement asynchrones

Les premiers systèmes de télé-enseignement qui ont été technologiquement possibles et pour lesquels de nombreux prototypes ont été réalisés sont des systèmes asynchrones. Ces systèmes permettent à des étudiants de consulter des cours à distance, et d'avoir recours à un enseignant lointain si nécessaire. Ces systèmes se composent généralement :

- de bases de données contenant des cours que les apprenants peuvent consulter à distance, en les téléchargeant de façon plus ou moins interactive sur leur ordinateur ;
- d'enseignants auxquels on peut poser des questions ;
- de réseaux de communication informatiques interconnectant les stations des étudiants et des enseignants avec les bases de données.

Le réseau américain K12 [MUR93] [HAN94] a ce type d'architecture. Il offre aux étudiants de très nombreux cours textuels stockés dans des bases de données réparties.

Les étudiants peuvent donc consulter individuellement ces cours, mais s'ils rencontrent des difficultés, une liste de professeurs spécialistes dans le domaine et leurs adresses électroniques leur est proposée. Les apprenants peuvent alors s'adresser à eux et leur soumettre leurs questions ; ils leur envoient un courrier électronique auquel le ou les enseignants répondent par un autre courrier électronique.

Une autre expérience similaire a vu le jour en Italie ; dans ce système, les communications se font par l'intermédiaire d'un outil de courrier électronique évolué (TEMPO) [POZ94] [POZ95] qui inclut dans le message le contexte de la question, de façon à éviter toute forme de quiproquo, ou de mauvaise compréhension entre les deux parties impliquées dans cette interaction.

Cependant, actuellement, le système de télé-enseignement asynchrone qui connaît le plus de succès et qui est en train de devenir un standard de fait utilise le "World Wide Web" (ou WWW ou W3). Au cours de l'année écoulée, ce système a connu une formidable explosion ; de très nombreux cours ont été écrits (en utilisant le langage HTML [DAG94]) [AND95][MAR95][NAK95][WOL95], et ont été mis à la disposition de la communauté Internet sur les serveurs W3 installés dans de très nombreux organismes mondiaux. Ainsi, l'apprenant a la possibilité de consulter les cours stockés sur les serveurs W3 par l'intermédiaire de logiciels de consultation tels Mosaic, Lynx ou Netscape ; de plus, le haut niveau d'interconnexion des serveurs W3 permet à l'étudiant d'avoir accès à tous les documents qui l'intéressent sur tous les serveurs W3 sans avoir à se soucier de la répartition des documents, la répartition lui étant transparente. En ce qui concerne les possibilités d'interaction avec des personnes distantes, ce système intègre deux solutions :

- la première consiste, de façon classique, à envoyer un e-mail à l'auteur du document que l'on est en train de consulter, les logiciels de consultation intégrant en général une interface conviviale pour le courrier électronique ;
- l'autre possibilité consiste à attacher au document des annotations auxquelles l'enseignant pourra ensuite répondre par e-mail, par d'autres annotations, ou en modifiant le document de façon à le rendre plus compréhensible au regard de la question posée.

Ces systèmes de télé-enseignement asynchrones ont un réel intérêt par rapport aux systèmes d'EAO, car ils offrent la possibilité de communiquer pour avoir des explications ou des points de vue de personnes différentes. Ils sont de plus massivement implantés dans le monde Internet et massivement utilisés au moins au niveau d'organismes professionnels. Toutefois, l'asynchronisme des communications réduit considérablement le niveau d'interactivité entre deux individus et l'efficacité de ces systèmes dans de très nombreux cas de formation est insuffisante.

1.2.2. Les systèmes de télé-enseignement synchrones

En réponse à ce problème, de nombreux efforts sont faits dans le domaine des systèmes de télé-enseignement synchrones. Ces systèmes ont souvent les mêmes possibilités que les systèmes asynchrones, i.e. ils permettent de télé-charger des cours depuis des bases de données distantes et de communiquer par courrier électronique ; ils intègrent également des procédés d'interaction synchrones temps réel comme des systèmes de téléconférence. Par exemple, FIRN (Florida Information Resource Network [BAR93b]) est un système qui permet des interactions synchrones en utilisant un mail électronique temps réel - une sorte de téléconférence textuelle. Ce système permet donc de créer une forme de dialogue entre les deux interlocuteurs, qui est impossible en utilisant le courrier électronique classique.

Dans la continuité de cet effort, de très nombreux outils ont été conçus pour améliorer la convivialité des communications et étendre les moyens d'interaction. Pour cela, il a fallu avoir recours aux technologies multimédias et de communication haut débit existantes, même si elles sont parfois encore un peu limitées. Ainsi, il est possible de citer :

- ISDN-IDL (Integrated Services Data Network - Interactive Distance Learning) [HGT92] et CO-LEARN [DER93][CRO94] qui améliorent FIRN en ajoutant des données graphiques et audio et en remplaçant le courrier électronique temps réel par une application d'audioconférence. Grâce à l'audioconférence (au dessus du RNIS), le niveau d'interactivité s'est considérablement amélioré en rendant le dialogue oral possible entre un enseignant et un enseigné par exemple ;

- "Electronic classroom" [ELL94] ou "ImagineDesk" [POZ94][POZ95] offrent la possibilité d'interagir sur un document partagé (partage d'écran ou espace de travail partagé) ;

- "Virtual classroom" [WIL94], "Virtual campus" [PAQ95], les travaux de [ESC95] ou TOPASE [OWE95b] sont aujourd'hui les systèmes qui envisagent de recréer le plus fidèlement possible, au travers de réseaux d'ordinateurs, les relations qui existent entre un enseignant et des enseignés dans une salle de classe. Pour cela, ils intègrent des communications vidéo, des systèmes de visioconférence, des tableaux de communication, tableaux électroniques, tableaux de dialogue, tableaux blancs, partage d'applications, etc. et envisagent d'utiliser des communications multipoints pour permettre une coopération efficace entre tous les participants aux cours.

Tous les systèmes de télé-enseignement évoqués dans cette partie sont maintenant classés dans le tableau V.1 qui montre l'évolution de ces systèmes selon les deux axes : introduction de technologies multimédias, et utilisation de moyens de communication.

Systèmes de télé-enseignement	Asynchrones	Synchrones
Textuels	K12 TEMPO	FIRN
Multimédias	W3	ISDN-IDL CO-LEARN Virtual Classroom Virtual Campus ImagineDesk TOPASE

Tableau V.1. Classification des systèmes de télé-enseignement

En fait, les difficultés majeures pour la réalisation de systèmes de télé-enseignement multimédias synchrones sont celles exposées au chapitre 1 ; ces difficultés concernent :

- l'introduction de la vidéo dans les cours et surtout au niveau des communications et de la synchronisation car elle requiert de la part du réseau non seulement beaucoup de bande passante, mais aussi des services temporels qui ne sont pas disponibles sur les supports de communication actuels ;
- le niveau d'interactivité entre deux interlocuteurs qui doit être élevé, ce qui nécessite d'avoir des supports de communication et de traitement respectant des contraintes temps réel ;
- l'introduction de communications multipoints pour permettre de travailler de façon efficace en groupe, et surtout de pouvoir définir, modéliser et reproduire les interactions entre les différents membres du groupe.

2. L'application VACBI

La société Airbus Training, d'Airbus Industries, a pour rôle d'assurer la formation des personnels (pilotes et agents de maintenance) intervenant sur les avions produits par le consortium européen. Ces cours sont spécialisés par type d'appareils et sont destinés à des personnels déjà qualifiés et qui devront travailler sur ces avions. Des cours ont été développés pour l'A300, l'A320 et maintenant pour l'A340.

2.1. L'application VACBI actuelle

Ces cours sont construits sur le système VACBI en utilisant les concepts de l'EAO basés sur l'utilisation du texte, du graphique, du son et de la vidéo (c'est un système d'EAO multimédia appartenant donc à la deuxième classe de la taxonomie présentée). Un pilote, un co-pilote ou un agent de maintenance qui viennent se perfectionner sur un avion, disposent des informations leur permettant d'acquies correctement toutes les procédures correspondant à leurs responsabilités et à leurs tâches dans le déroulement de leur travail. Le pilote apprendra par exemple à manipuler la sortie et la rentrée normale du train d'atterrissage, et à réagir face à des situations anormales, prises en compte dans le module *train d'atterrissage* du cours. Pour cela les médias qui apparaissent sont :

- des graphiques pour représenter les instruments et les panneaux de commande concernés par les procédures étudiées ;
- des textes pour indiquer les opérations à réaliser et d'éventuelles erreurs ;
- des messages sonores pour guider le stagiaire ;
- des séquences vidéo pour montrer l'effet réel des commandes exécutées.

Toutes ces informations sont présentées de façon séquentielle, et en utilisant deux écrans : l'écran de la station sur lequel s'exécute le logiciel VACBI et un moniteur de télévision connecté à un lecteur de vidéodisques piloté via une RS232 par la station. La station actuelle est un PC connecté à un réseau Ethernet qui permet de télécharger les cours (hormis la vidéo) d'un serveur d'application (figure V.1).

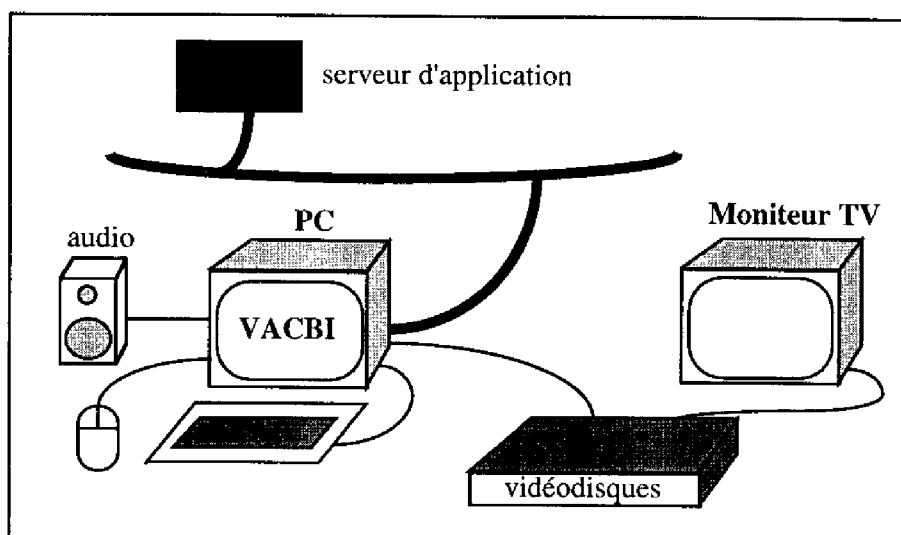


Figure V.1. Un poste d'EAO VACBI

Les cours sont dispensés dans les trois centres de formation d'Airbus Training (dont 250 postes à Toulouse et 50 postes à Miami), mais ils sont également accessibles sur des postes autonomes dans des escales ou au siège des compagnies aériennes clientes (plusieurs centaines de postes répartis dans le monde). Actuellement, chaque élève suit une formation devant un poste de travail autonome. Le cours du personnel navigant comporte 54 heures et se déroule pendant deux semaines. Ce cours est suivi de séances sur simulateur de vol (pendant 3 semaines). Avant cette technique, une compagnie aérienne mettait environ trois mois pour former son personnel navigant sur les nouveaux appareils acquis.

Chaque cours comporte un instructeur et plusieurs stagiaires : un cours pilote est suivi par un instructeur pour 20 élèves (1 élève par poste), tandis qu'un cours

maintenance est suivi par un instructeur pour 16 élèves (2 élèves par poste). Chaque séance se décompose en 3 phases :

- une phase de briefing au cours de laquelle l'instructeur présente l'objet du cours et les modules qui devront être étudiés. Il peut également prodiguer quelques conseils par rapport à son expérience de ces modules, et notamment indiquer le meilleur ordre de parcours des différents modules. Enfin, une discussion peut avoir lieu entre les stagiaires et l'instructeur, au cours de laquelle ce dernier pourra avoir à répondre aux premières questions découlant de sa présentation du cours ;
- une phase d'auto-apprentissage au cours de laquelle les stagiaires travaillent seuls ou par deux sur leur poste VACBI. Pendant cette phase, l'instructeur reste dans la salle, et peut répondre à d'éventuelles questions que lui poseraient des stagiaires. Enfin, l'instructeur surveille le bon déroulement de la séance d'auto-apprentissage pour s'assurer que tous les étudiants parviennent à réaliser le travail qui leur a été demandé ;
- enfin, une phase de debriefing clôture le cours. Cette phase est une réunion au cours de laquelle les stagiaires et l'instructeur débattent des difficultés rencontrées ; les stagiaires peuvent poser leurs dernières questions, tandis que l'instructeur peut utiliser cette période pour clarifier d'autres points et faire le bilan du travail effectué.

Toutefois, ce type de formation présente quelques inconvénients :

- les frais de voyage des stagiaires et des instructeurs accroissent le coût de la formation des personnels navigants et de maintenance ;
- lorsque des modifications ou des nouveautés apparaissent sur les cours, elles sont difficiles à prendre en compte (en particulier en ce qui concerne les séquences vidéo analogiques : tous les vidéodisques doivent être changés sur tous les postes VACBI, ce qui ne peut se faire par le réseau puisque seules les données numérisées peuvent être téléchargées) ;
- enfin, une fois la formation finie, il est difficile d'informer les stagiaires des nouveautés sur l'avion, et d'assurer un suivi du cours.

2.2. Les extensions proposées pour l'application VACBI

Aussi, nous avons proposé des extensions potentielles pour tenter d'améliorer la version actuelle de VACBI. Ces extensions se divisent en deux groupes : les extensions locales et les extensions distribuées.

Parmi les extensions locales, nous avons proposé (par l'intermédiaire d'une maquette) de modifier le système d'EAO, de façon à ne manipuler que des données numérisées affichées sur un écran unique et de façon synchronisée [FAN96] [CIC95] [BAU96a]. La numérisation de tous les médias (en particulier la digitalisation de la vidéo) semble très intéressante pour la gestion des différentes version du cours et leur téléchargement. De plus, l'intégration de la vidéo sur le même écran que les textes et graphiques améliorerait l'ergonomie du système d'EAO : il est plus facile pour un étudiant de regarder sur un seul écran plutôt que deux, et il peut donc être plus attentif à tous les panneaux de commande du cours. Enfin, la synchronisation entre les médias permettrait un meilleur rendu de l'application car tous les médias sont toujours présentés au même niveau d'avancement : les stagiaires pourraient donc voir en temps réel les effets de leurs actions conjointement sur l'avion et les instruments de bord.

Le second type d'extensions vise à transformer le système d'EAO multimédia VACBI en un système de télé-enseignement multimédia synchrone (appartenant à la quatrième classe de la taxonomie présentée en 1). Pour cela, il est nécessaire de développer des outils qui permettent de recréer au travers de stations de travail et de réseaux de communication le même niveau d'interactivité qui existe entre un instructeur et des stagiaires dans une salle de formation VACBI. Les outils proposés sont des outils de visioconférence de bureau qui permettent aux utilisateurs de communiquer par la voix, la vidéo et un tableau électronique de dialogue qui remplace le tableau noir traditionnel des salles de cours. Un tel système permettrait des formations beaucoup plus flexibles, puisque les stagiaires pourraient étudier lorsqu'ils ont du temps libre,

plutôt que 8 heures par jour pendant deux semaines. De plus, ce système permettrait aux stagiaires de regarder régulièrement le cours pour perfectionner leur connaissance de l'avion ou pour prendre connaissance des dernières modifications effectuées.

3. Etude des interactions entre instructeur et stagiaires

Pour pouvoir réaliser un environnement de téléformation il a été décidé d'étudier les interactions qui existent dans une salle classique entre l'instructeur et les stagiaires et de les reproduire dans notre environnement virtuel [KAR94b].

Dans une salle de cours classique, l'enseignant et les élèves sont regroupés géographiquement en un même lieu. Différents types d'interactions permettent à l'instructeur de diffuser ses connaissances aux stagiaires, et lui permettent également de s'assurer que celles-ci ont été correctement assimilées.

Dans un environnement distribué, chaque participant va devoir disposer d'une station de travail connectée à un réseau de communication. Les interactions directes observées dans une salle de cours devront alors être possibles par le biais d'outils logiciels mis à la disposition des participants. Il est donc nécessaire, dans un premier temps, d'identifier les interactions instructeur/stagiaire(s) dans un contexte de briefing/debriefing, et dans un contexte de cours Airbus. Celles-ci devront ensuite être modélisées dans le but de les implémenter dans des outils.

3.1. Interactions instructeur/stagiaire(s) dans un contexte de briefing/debriefing

Dans un contexte de briefing, l'instructeur détient les informations qu'il diffuse aux stagiaires sous la forme de transparents projetés sur un écran, ou sous forme de notes (textuelles et graphiques) écrites au tableau.

Pendant le déroulement du cours, un élève peut poser une question orale à l'enseignant, celle-ci étant entendue par l'ensemble de la classe.

La question posée peut également nécessiter la désignation d'une zone particulière à l'écran ou au tableau, cette désignation devant être vue par l'ensemble des autres stagiaires. La question posée peut aussi nécessiter une action de la part du stagiaire, comme par exemple se déplacer au tableau pour tracer un schéma. Cette action doit également être vue de tous.

Enfin, la réponse fournie par l'instructeur doit être entendue par l'ensemble de la classe [KAR94b] [OWE96f] [OWE96g].

Un graphe nous permet de représenter ces échanges d'informations. Il est composé d'un noeud-instructeur, et de N noeuds-stagiaires. Des flèches indiquent le sens de diffusion des informations. Nous considérons quatre types d'informations échangées :

- le discours (les paroles et la gestuelle) ;
- le support d'information (transparents, notes au tableau ou sur papier, système électronique de présentation) ;
- les désignations de zones du support d'information ;
- des actions sur un support d'information.

Les quatre types de flèches représentant les différents types d'informations échangées, ne sont pas obligatoirement utilisés pour une situation donnée.

On peut alors représenter les graphes modélisant les 3 situations décrites plus haut :

- l'instructeur présente le cours, les modules qui devront être parcourus, l'ordre de parcours, etc. (Figure V.2). Sur le graphe, le noeud-instructeur a une position centrale pour la diffusion du support d'informations ;

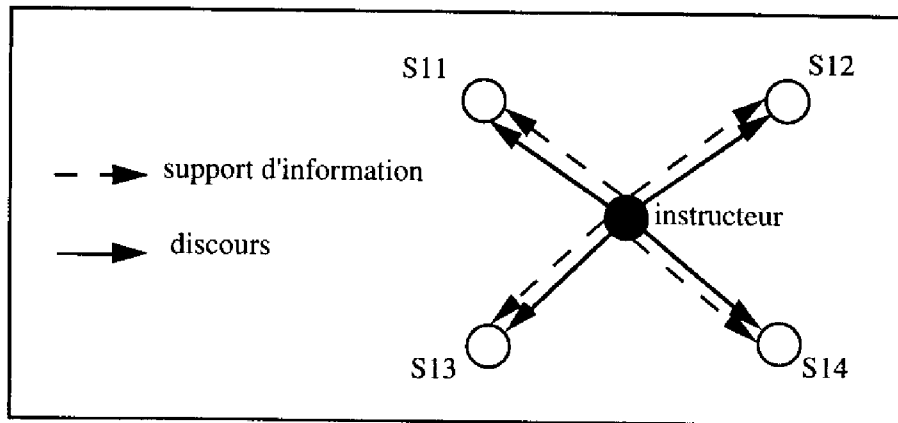


Figure V.2. Graphe de coopération en phase de briefing : l'instructeur présente le cours

• un stagiaire pose une question orale, sans désignation sur le support d'information utilisé par l'instructeur (Figure V.3). Le support d'information est diffusé par le noeud-instructeur, le discours est diffusé par le noeud-S11 ;

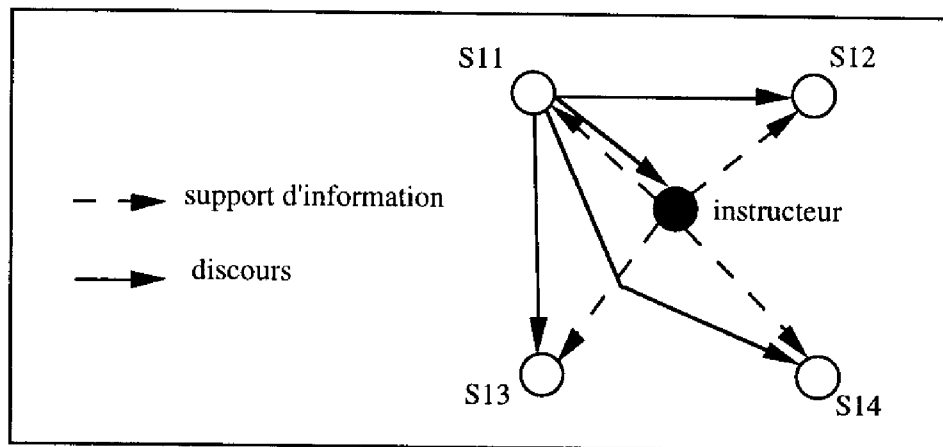


Figure V.3. Graphe de coopération en phase de briefing : un stagiaire pose une question sans référencer le support de cours

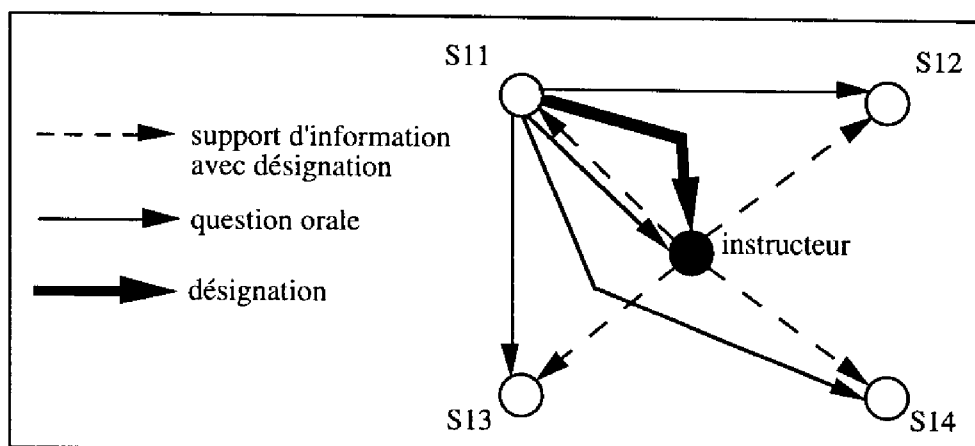


Figure V.4. Graphe de coopération en phase de briefing : un stagiaire pose une question en référençant le support de cours

- un stagiaire pose une question orale et montre un point sur le support d'information de l'instructeur (Figure V.4). Le support d'information est diffusé par le noeud-instructeur. Ce support est enrichi par la désignation ou la commande appliquée par le noeud-S11. La désignation ou la commande est donc diffusée par le noeud-S11 vers le noeud-instructeur, le noeud-instructeur se chargeant de la diffusion aux noeuds-stagiaires du support "enrichi".

Si l'on veut conserver un même type de fonctionnement dans un environnement distribué, les contraintes suivantes seront à prendre en compte :

- tous les stagiaires doivent voir le même support produit par l'instructeur : transparent, séquence d'une application d'EAO (VACBI), document hypertexte, objet filmé ;
- tous les stagiaires doivent entendre le discours de l'instructeur synchronisé [OWE95b] avec ce qu'il montre, et avec son image ;
- un stagiaire peut poser des questions que l'instructeur et les autres stagiaires doivent pouvoir entendre ;
- un stagiaire peut montrer un point précis sur son écran que tous (instructeur et autres stagiaires) doivent pouvoir voir ;
- un stagiaire doit pouvoir agir sur l'application de l'instructeur, lorsqu'il pose une question : il est cependant nécessaire de restreindre les droits d'accès d'un stagiaire sur la station de l'instructeur. L'instructeur doit toujours être prioritaire sur sa station ;
- l'application (outil de présentation de supports, par exemple) est résidente chez l'instructeur uniquement ;
- un stagiaire ou un instructeur doit pouvoir prendre des notes ;
- l'instructeur ou un stagiaire peut prendre la parole [KAR94b].

3.2. Interactions instructeur/stagiaire(s) dans un contexte d'auto-apprentissage VACBI

Dans un contexte d'auto-apprentissage VACBI, les stagiaires ont un rôle plus actif : chacun doit parcourir un ou plusieurs modules de cours.

Chaque stagiaire travaille donc dans un environnement qui lui est propre. L'état d'avancement dans le module n'est pas identique pour tous : chaque stagiaire détient des informations personnelles relatives au module qu'il étudie. L'instructeur, quant à lui doit pouvoir "passer" derrière chaque stagiaire pour juger de son avancement, et lui apporter éventuellement son aide. Cette aide de l'instructeur peut être apportée par le biais d'un dialogue instructeur/stagiaire, mais aussi par le biais d'une intervention plus directe de l'instructeur : ce dernier peut, par exemple, utiliser la souris de la station de travail de l'élève pour effectuer une commande (click bouton, déplacement d'une fenêtre, ...).

On peut également observer une situation dans laquelle l'instructeur demande l'attention de l'ensemble des stagiaires pour diffuser une information qu'il juge importante : il y a dans ce cas basculement d'un état d'aide personnalisée, à un état que l'on pourrait qualifier "d'aide généralisée".

On peut, comme dans le contexte de briefing, représenter sous forme de graphes, les échanges d'informations ayant lieu dans les situations d'auto-apprentissage VACBI décrites ci-dessus :

- L'aide personnalisée à un stagiaire (Figure V.5) : trois types d'informations sont échangées dans cette situation. Il y a le dialogue entre l'instructeur et le stagiaire ayant demandé de l'aide ; l'instructeur voit le support d'information utilisé par l'élève, et peut agir dessus. Le support d'information appartient au noeud-stagiaire, mais les informations supplémentaires du type désignation et commande peuvent provenir du noeud-instructeur. Dans cette situation, les autres stagiaires ne voient pas les échanges d'informations entre le noeud-instructeur et le noeud-S11 ;

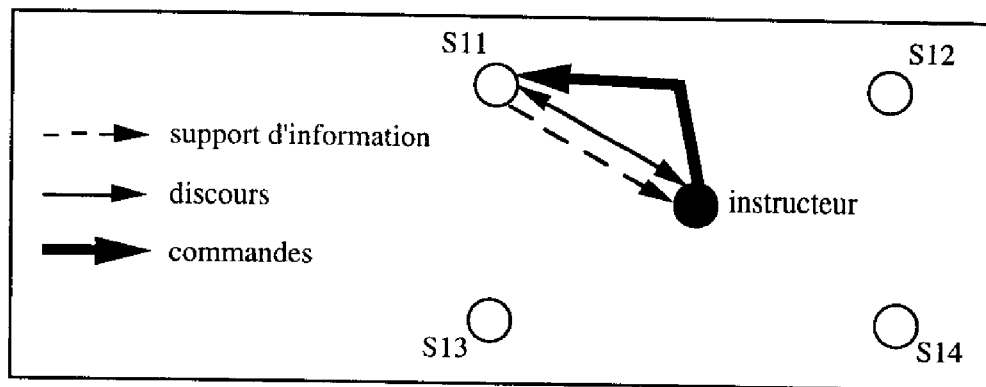


Figure V.5. Graphe de coopération en phase d'auto-apprentissage : l'aide personnalisée

• L'aide généralisée avec diffusion d'informations "stagiaire" (Figure V.6). Suite à une question de S11, l'instructeur souhaite faire profiter l'ensemble des stagiaires de sa réponse. Dans cette situation deux sources émettent des informations vers l'ensemble des participants : le discours de l'instructeur est diffusé à partir du noeud-instructeur vers l'ensemble des N noeuds-stagiaires, le support d'information de l'élève S11 ayant initialement demandé de l'aide est diffusé au noeud-instructeur et aux N-1 autres noeuds-stagiaires à partir du noeud-S11. Le support d'information appartenant au noeud-S11 peut être enrichi par une désignation ou une commande provenant du noeud-instructeur. Dans ce cas, nous aurons un arc orienté supplémentaire de type désignation ou commande du noeud-instructeur vers le noeud-S11.

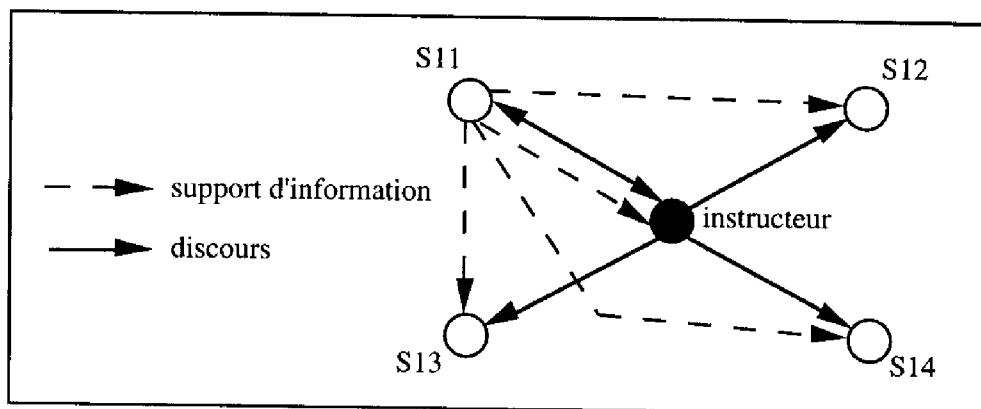


Figure V.6. Graphe de coopération en phase d'auto-apprentissage : l'aide généralisée

• L'aide généralisée avec diffusion d'informations "instructeur" : suite à une question de S11, l'instructeur souhaite faire profiter l'ensemble des stagiaires de sa réponse : nous nous retrouvons dans la situation présentée sur la Figure V.1. Le discours de l'instructeur et son document sur lequel s'appuie sa réponse sont diffusés à tous à partir du noeud-instructeur. Nous retrouvons donc le noeud-instructeur dans une position centrale de diffusion.

On peut donc définir l'aide généralisée comme un état dans lequel l'instructeur assure un rôle central pour la diffusion d'informations en provenance d'applications appartenant à un stagiaire ou à lui-même (fonctionnements représentés par les figures V.2, V.3, V.4 et V.6).

Pour répondre à ces trois situations, les fonctionnalités pour chaque type de participant sont alors :
Pour un stagiaire de pouvoir :

- demander de l'aide ;
- montrer à distance ;
- commander à distance avec restriction des droits d'accès ;
- dialoguer avec l'instructeur et le voir.

Pour l'instructeur de pouvoir :

- commander à distance sans restriction de droits d'accès ;
- assurer le suivi d'un stagiaire (aide sans commande) ;
- dialoguer avec un stagiaire et le voir ;
- superviser l'ensemble des stagiaires ;
- basculer en "aide généralisée".

D'autres fonctionnalités annexes peuvent être offertes, sans être spécifiques à l'une de ces 3 étapes :

pour un stagiaire :

- consulter la liste des autres stagiaires participant à la formation ;
- appeler un stagiaire ;
- répondre à un stagiaire ;
- constituer une fiche de renseignements ;
- consulter la partie non confidentielle d'une fiche de renseignements d'un stagiaire.

pour l'instructeur :

- construire ses supports de cours (transparents) ;
- consulter une fiche de renseignements ;
- consulter une fiche de suivi de cours.

Cependant, dans ce qui suit, ces fonctionnalités secondaires ne seront pas approfondies, ce papier s'intéressant exclusivement aux interactions fondamentales de la téléformation synchrone.

4. Les outils

La partie précédente a modélisé les différents types d'interactions qui doivent apparaître dans le cadre d'une téléformation. Dans les modèles précédents, plusieurs types de flux de données ou de contrôles apparaissent : des données audio et vidéo de dialogue, des supports de cours, des informations de pointage et des instructions de commandes à distance. En fait, ces interactions peuvent se classer en deux familles complémentaires : les données du dialogue oral entre personnes et les données propres au support de cours. Deux outils ont donc été conçus pour s'occuper en exclusivité de l'une de ces familles d'interactions : un outil de visioconférence pour le dialogue entre les individus impliqués dans le cours (cet outil semble incontournable chaque fois que la réalisation d'une application distribuée coopérative synchrone est envisagée), et un outil de support de cours - appelé "*tableau de dialogue*" - qui permet la diffusion des supports de cours, mais aussi le partage d'applications et qui peut également servir de tableau blanc.

4.1. L'outil de visioconférence

Cet outil permet à plusieurs utilisateurs de communiquer par l'intermédiaire de la voix et de l'image, et de reproduire le dialogue qui existe entre les participants d'un cours dans une classe normale. Pour cela, deux formes de l'outil de visioconférence doivent être considérées : un outil de visioconférence point à point et un outil de visioconférence multipoints.

La visioconférence point à point permet à deux utilisateurs de communiquer sans que les autres participants du cours ne soient impliqués. Cet outil pourrait être utilisé

dans le cadre d'auto-apprentissage VACBI pour lesquels chaque stagiaire travaille sur un module de façon indépendante, et le dialogue doit donc être privé entre le stagiaire concerné et l'instructeur. Pour une bonne ergonomie de l'outil, des mécanismes de synchronisation multimédia ont été implémentés de telle sorte que les mouvements des lèvres correspondent aux mots prononcés. Cette synchronisation a été implémentée dans TSVS en utilisant des estampilles temporelles [DIA94a] [OWE95a] [PAR94] (cf. chapitre 1) ou en implémentant un moteur de présentation de scénarios multimédias modélisés à l'aide d'un TSPN dans PNSVS [OWE95c] [OWE96c] (cf. chapitre 2). Néanmoins, les communications audio / vidéo point à point peuvent aussi être réalisées en utilisant l'outil de visioconférence coopératif avec un groupe composé exclusivement de deux utilisateurs.

La visioconférence multipoints pour N participants appartenant à un groupe structuré est toutefois obligatoire en phase de briefing/debriefing, ou en phase d'aide généralisée d'un cours. Dans ces cas, les informations sont données à l'ensemble de la classe, et le dialogue concerne tous les participants. Dans N-TSVS [OWE96a] [OWE96b] [OWE96d] (cf. chapitre IV), la synchronisation multimédia entre les flux audio et vidéo est réalisée en utilisant des estampilles globales. Cette visioconférence permet également des entrées et sorties dynamiques du groupe des participants au cours (participation dynamique) [DIA93d][VIL95]. Ceci autorise donc les arrivées tardives ou les départs anticipés pour certains participants. En fait, les notions de groupe auto-régulés et de communication de groupes sont essentielles dans cette application.

4.2. Le tableau de dialogue

Le second outil qui a été développé est appelé tableau de dialogue. Cet outil a de multiples fonctionnalités : il permet par exemple de transmettre sous forme vidéo n'importe quel objet ou document papier dont l'image est capturée par une caméra de table. Il permet également de transmettre des documents multimédias (intégrés dans des fenêtres), sert de tableau blanc ainsi que d'outil de partage d'applications (en utilisant des mécanismes de télépointage) [BAU96b].

4.2.1. Les fonctionnalités

En fait, en se ramenant aux situations modélisées dans la partie 3, il apparaît plusieurs modes de fonctionnement du tableau de dialogue :

- Dans le cadre d'un briefing, l'instructeur détient, sur sa station, les informations qu'il souhaite communiquer aux stagiaires. De ce fait, la station de l'instructeur sera un point central du cours. C'est le cas par exemple lorsque l'enseignant présente ou montre son cours aux élèves avec les informations de pointage (cf. figure V.2). Cette situation correspond donc à une diffusion de l'information de la station de l'instructeur vers les stations des stagiaires : cette fonctionnalité est appelée ShowN (car l'instructeur montre un document aux N élèves de son cours).

Cependant, un stagiaire peut avoir à référer de façon explicite le document diffusé pour formuler sa question. Il est également envisageable que le stagiaire ait besoin d'agir sur le document en question (dans le cas où il serait interactif) pour bien montrer le point qu'il ne comprend pas. Dans ce cas, il peut avoir à prendre le contrôle de l'application de l'instructeur dont il reçoit l'image (cf. figure V.4); cette image, ainsi que le pointeur du stagiaire seront alors diffusés vers le reste de la classe de sorte que toute la classe puisse disposer du contexte de la question posée. Cette fonctionnalité s'appelle HelpShowN, car le stagiaire contrôle l'application de l'instructeur, comme le ferait ce dernier s'il avait à aider un élève, et les informations sont ensuite diffusées à toute la classe. Toutefois, le contrôle du stagiaire sur l'application de l'enseignant n'est pas total, et ce dernier, par son rôle hiérarchiquement prépondérant peut reprendre le contrôle de son application quand bon lui semble : pour cela, il est possible, par exemple, d'attribuer le contrôle du pointeur de la souris à celui qui la bouge, et ce avec priorité à

l'enseignant. Ainsi, il suffit à l'instructeur de déplacer sa souris pour reprendre le contrôle de son application.

- Dans le cadre d'une phase d'auto-apprentissage de type VACBI, les informations sont détenues par les stagiaires qui traitent localement un exercice. Dans ce cas, les communications sont la plupart du temps privées (point à point) entre l'instructeur et un stagiaire. Les fonctionnalités de base permettent donc à un étudiant de montrer son document ainsi que les actions qu'il effectue (par transmission des informations de pointage) : cette fonctionnalité s'appelle Show1. De même, pour répondre à la question du stagiaire, l'instructeur peut avoir à prendre le contrôle de l'application de l'élève (cf. figure V.5), pour lui montrer ce qu'il doit faire. Dans ce cas, la souris du stagiaire est bloquée et ne peut être débloquée que par la fin de l'intervention de l'instructeur (contrairement à ce qui se passe quand l'élève prend le contrôle sur la station du professeur) : cette fonctionnalité s'appelle Help1.

Il est également possible d'utiliser les fonctionnalités HelpShowN et ShowN qui ont été définies pour les phases de briefing/debriefing. Ces fonctionnalités prennent toute leur signification dans le contexte d'une aide généralisée (cf. figure V.6) où, suite à un dialogue privé, l'instructeur décide que le problème qui lui est soumis et la solution qu'il va apporter méritent d'être présentés à l'ensemble de la classe.

4.2.2. Les problèmes de synchronisation

Quatre fonctionnalités ont donc été identifiées pour le tableau de dialogue, avec deux variantes lorsqu'il y a une prise de contrôle à distance, suivant l'identité de la personne qui initie cette prise de contrôle. Cependant, dans ces 4 types d'interactions, 3 sortes d'informations transitent : les informations vidéo (image du document présenté), les informations de pointage, et les commandes à distance (voir partie 3). Aussi, il faut pouvoir synchroniser tous ces flux d'information et de commande de sorte que les informations de pointage pointent bien sur les informations à pointer, par exemple.

- Dans le cas du Show1 et du ShowN, les informations vidéo et de pointage transitent en parallèle. Aussi, il est important que sur la (les) station(s) réceptrice(s) le pointeur transmis soit présenté avec l'image vidéo sur laquelle il doit pointer. Pour ceci, nous pouvons utiliser les estampilles déjà utilisées dans TSVS ou les TSPN utilisés dans PNSVS ; pour les raisons déjà évoquées ces deux techniques garantissent les synchronisations intra et inter-flux².

- Pour le Help1, le problème est entièrement différent, puisque les informations vidéo vont dans un sens alors que les informations de commande à distance vont dans le sens opposé. De plus, celui qui commande l'application distante ne peut intervenir que lorsqu'il en reçoit l'image vidéo, et son action ne pourra être prise en compte que lorsque sa commande sera parvenue à la station centrale et initiatrice (pour les données vidéo). Aussi, il ne faut pas que l'application contrôlée à distance évolue tant que les instructions de commande n'auront pas été reçues en retour. Cette synchronisation entre l'information pointée et son pointeur se fait donc par un blocage à la source.

- Pour le HelpShowN, la synchronisation est basée sur les deux principes précédents, à savoir des estampilles ou le moteur TSPN pour assurer la synchronisation sur les stations réceptrices pures, et le blocage pour les stations impliquées dans la prise de contrôle à distance.

4.2.3. Principes de fonctionnement du tableau de dialogue

Dans le tableau de dialogue, les applications informatiques, ou plus exactement leurs images, sont transmises sous forme vidéo, i.e. l'image vidéo d'une fenêtre est capturée sous forme d'une matrice de pixels (bit-map) compressée, transmise et présentée sur la(les) station(s) distante(s). Ce choix est peu fréquent, les autres essais ayant plutôt eu tendance à transmettre le code des applications ou la liste des événements du système

²Dans un souci de simplification, les implémentations actuelles utilisent les estampilles.

de multifenêtrage [CHU94]. La suite va donc présenter les avantages et inconvénients de cette approche.

Inconvénients :

Cette approche représente une perte d'information au sens informatique. En effet, une zone de texte dans l'application initiale est perdue, puisque, sémantiquement, sur l'image reçue, l'information textuelle disparaît, l'application réceptrice ne pouvant voir que de la vidéo, ce qui appauvrit donc l'image de l'application reçue.

De plus, transmettre un texte sous forme vidéo induit un débit plus important sur le réseau.

Enfin, même si la vidéo permet de transmettre des données visuelles comme du texte, du graphique et de la vidéo, la vidéo ne peut contenir les informations sonores de l'application.

Avantages :

Toutefois, cette approche présente l'avantage d'être très simple, et ne nécessite pas, pour partager une application de connaître la sémantique constitutionnelle de cette application. De plus, même si informatiquement l'image reçue représente une perte d'information par rapport à l'application initiale, l'utilisateur peut lire dans l'image les informations textuelles ou graphiques dont il a besoin, ce qui justifie pleinement ce choix, d'autant plus qu'il permet à la fois d'utiliser le tableau de dialogue comme un espace de travail partagé, et comme un outil de partage d'application.

De plus, le tableau de dialogue permet de montrer à distance non seulement des objets numériques (fenêtres d'une application), mais aussi des documents papiers et des objets physiques (par exemple des pièces de l'avion dans le cadre de la formation des agents de maintenance) qui peuvent être capturés grâce à la caméra de table qui sert d'entrée au tableau de dialogue, et ce grâce à la transmission entièrement vidéo du logiciel.

La transmission entièrement vidéo de l'image de l'application partagée induit également plus de sécurité, car l'application en elle-même n'est pas transmise, et la vidéo seule ne permet pas de reconstituer l'application originelle qui est elle-même complètement multimédia (il n'y a pas transmission des données de l'application partagée vers l'extérieur).

Le principal avantage de l'approche "tout vidéo" (par rapport à la transmission des événements du système de multi-fenêtrage), dans le cadre de la téléformation Airbus, est incontestablement le gain en bande passante sur le réseau. En effet, le système d'ÉAO d'Airbus comporte un très grand nombre de séquences vidéo (presque tous les modules du cours en présentent) ; or en utilisant la transmission des événements du système de multi-fenêtrage (X), les données vidéo sont transmises sous forme de matrices de pixels ; le débit engendré est donc très important car les communications sous X se font avec des données non compressées. Ainsi, avec notre approche "tout vidéo", l'image de l'application est compressée au format JPEG, ce qui permet de réduire substantiellement le volume des données à transmettre.

5. Intégration et interface

La partie précédente vient de présenter les fonctionnalités des deux outils qui vont nous permettre de réaliser la base d'un environnement de téléformation synchrone émulant au mieux, dans un environnement distribué, les interactions traditionnelles entre stagiaires et instructeur dans une salle de cours. Ces deux outils sont très différents tant dans leur rôle que dans leur conception. Aussi, pour créer l'environnement de téléformation, il va falloir intégrer ces deux outils (la visioconférence et le tableau de dialogue), de telle sorte qu'ils fonctionnent ensemble, de façon complémentaire, et sans dérive. De même, il faut pouvoir définir une

interface, autant pour l'instructeur que pour les stagiaires, qui permette à chacun des participants au cours d'accéder facilement aux informations du cours et qui permette aisément d'entrer en contact et d'interagir avec les autres membres du groupe.

5.1. Intégration

L'intégration de ces deux outils de base dans l'environnement de téléformation doit respecter deux contraintes :

- Ils doivent fonctionner ensemble et de façon complémentaire (comme représenté sur les figures V.1 à V.6 de la partie 2). Cependant ce point se résoud de lui-même car les deux outils ont des fonctionnalités totalement complémentaires : la visioconférence s'occupe du dialogue audio/vidéo entre les participants alors que le tableau de dialogue gère toutes les autres interactions liées aux supports de cours échangés entre utilisateurs.

- Cependant, il faut veiller à ce qu'aucun décalage n'apparaisse. En effet, de part leur complémentarité, il existe une forte corrélation entre les données de la visioconférence et celles du tableau de dialogue, et il est important que cette corrélation ne soit pas détruite par la séparation des outils. Aussi, il est important de synchroniser les deux outils entre eux. La synchronisation a déjà été introduite de façon indépendante dans chaque outil et notamment en utilisant des estampilles temporelles ou des mécanismes de présentation conduits par des TSPNs.

Dans le cas de l'utilisation d'estampilles, leur intérêt est double puisqu'elles permettent la synchronisation des différents flux de chaque application, mais également la synchronisation inter-applications. Dans chaque application, les données multimédias sont datées en fonction d'une horloge, et sont synchronisées en fonction de cette base temporelle. De ce fait, comme les deux outils sont synchronisés individuellement en respectant la même horloge, ils sont donc synchronisés l'un par rapport à l'autre : c'est une synchronisation basée sur le paradigme de la ligne temporelle.

De même, dans les outils synchronisés par un moteur TSPN, il existe un mécanisme de contrôle du retard de présentation de bout en bout qui s'effectue par un contrôle au niveau du remplissage des tampons de réception [OWE95c]. De ce fait, si le retard de présentation est identique pour chaque application, la synchronisation inter-applications sera réalisée.

Cette synchronisation entre les deux outils devant fonctionner en parallèle, étant obtenue, il ne reste plus pour créer un environnement de téléformation qu'à définir une interface qui permette d'activer ces deux outils simultanément et de façon logique, tout en masquant la séparation existant entre ces outils au moment de leur conception. Ces deux outils doivent donc apparaître comme un outil unique à l'utilisateur [OWE96f].

5.2. Interfaces

Dans cet environnement de téléformation synchrone, deux types d'utilisateurs existent - les instructeurs et les stagiaires - et il existe donc deux interfaces pour les postes du système suivant que leur utilisateur appartient à l'une ou l'autre de ces deux catégories.

5.2.1. Interface du poste de l'instructeur

L'interface définie pour le poste de l'instructeur est représentée sur la figure V.7. En fait, 3 outils doivent être commandés par l'instructeur, et leurs "tableaux de commande" ont été intégrés dans cette interface.

Tout d'abord, l'enseignant doit gérer le groupe des participants au cours, et pour cela, il a à sa disposition un gestionnaire de groupe. Ce gestionnaire sert à informer l'instructeur de toutes les requêtes d'entrée et sortie du cours formulées par les stagiaires. L'enseignant est maître de sa décision, et dispose donc de 4 boutons :

- entrée_acceptée et entrée_refusée s'il accepte ou refuse la requête d'entrée d'un étudiant ;
- sortie_acceptée et sortie_refusée s'il accepte ou refuse le départ d'un des participants au cours ;
- fin_du_cours pour signaler à tous la fin du cours.

La seconde fenêtre qui apparaît sur le poste de l'instructeur est celle servant à piloter le tableau de dialogue. Elle comporte 3 parties :

- un espace de travail qui sert au transfert des informations vidéo du tableau de dialogue ;
- une zone indiquant le stagiaire avec lequel l'instructeur est en train de dialoguer ;
- une zone de boutons de commande qui permet d'accéder aux différentes fonctionnalités du tableau de dialogue, à savoir Help1, Show1, etc. ...

La troisième partie de l'écran est réservée à l'application de visioconférence. Une fenêtre de contrôle vidéo apparaît à l'écran pour chacun des stagiaires, ainsi qu'une fenêtre vidéo pour l'instructeur lui même. Elle a pour but de l'aider à se positionner par rapport à la caméra, vérifier la qualité de l'image, etc. ... De plus, à chaque fenêtre vidéo d'un stagiaire est associé un ensemble de boutons :

- un bouton permettant de démarrer ou stopper la partie audio associée à un stagiaire, afin de régler les problèmes de brouhaha lorsque le code de bonne conduite n'est pas respecté (cf. chapitre 4). Cependant, ce bouton n'est nécessaire que dans la mesure où le matériel audio utilisé aujourd'hui ne peut pas traiter le mixage de signaux multiples ;
- un bouton permettant d'accéder à une fiche du stagiaire qui peut par exemple décrire son cursus, ou ses formations antérieures ;
- un bouton qui permet à l'instructeur d'accéder au tableau de dialogue de ce stagiaire, pour superviser son travail, ou lorsque ce dernier fait appel à lui ;
- un voyant qui s'allume lorsque le stagiaire a une question à poser.

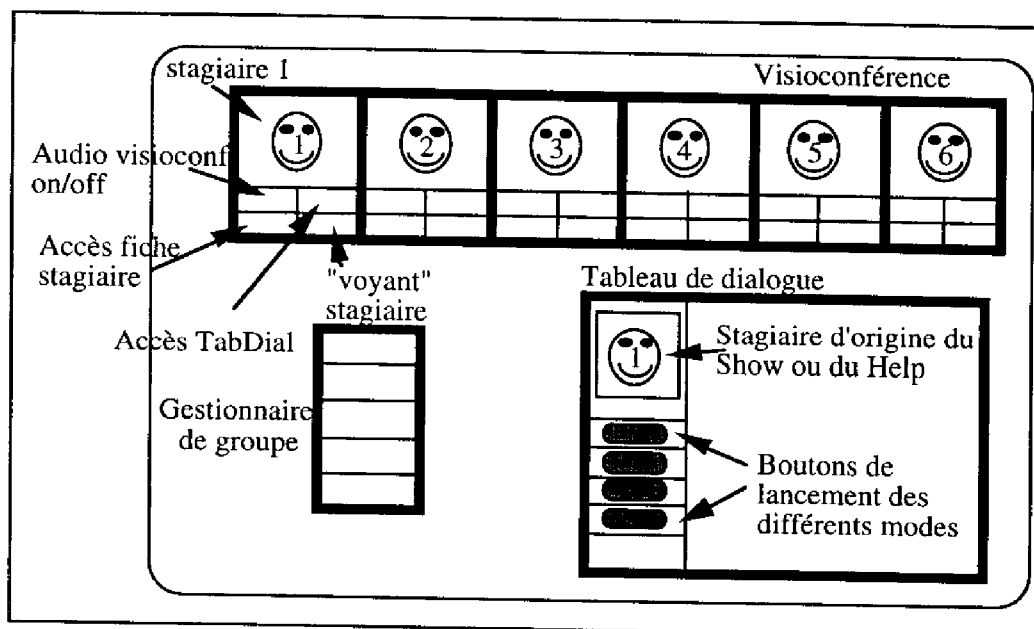


Figure V.7. L'interface du poste instructeur

5.2.2. Interface du poste d'un stagiaire

Le poste des stagiaires est représenté sur la figure V.8. Il ressemble beaucoup à celui de l'instructeur. Toutefois, la partie concernant la gestion du groupe disparaît, car seul l'instructeur décide des entrées et des sorties du groupe. La partie tableau de dialogue est similaire à celle qui a été définie pour le poste de l'instructeur. Enfin, la partie visioconférence demeure, le stagiaire pouvant voir et entendre l'ensemble des autres participants au cours. Toutefois, il ne dispose que d'un bouton pour appeler l'instructeur.

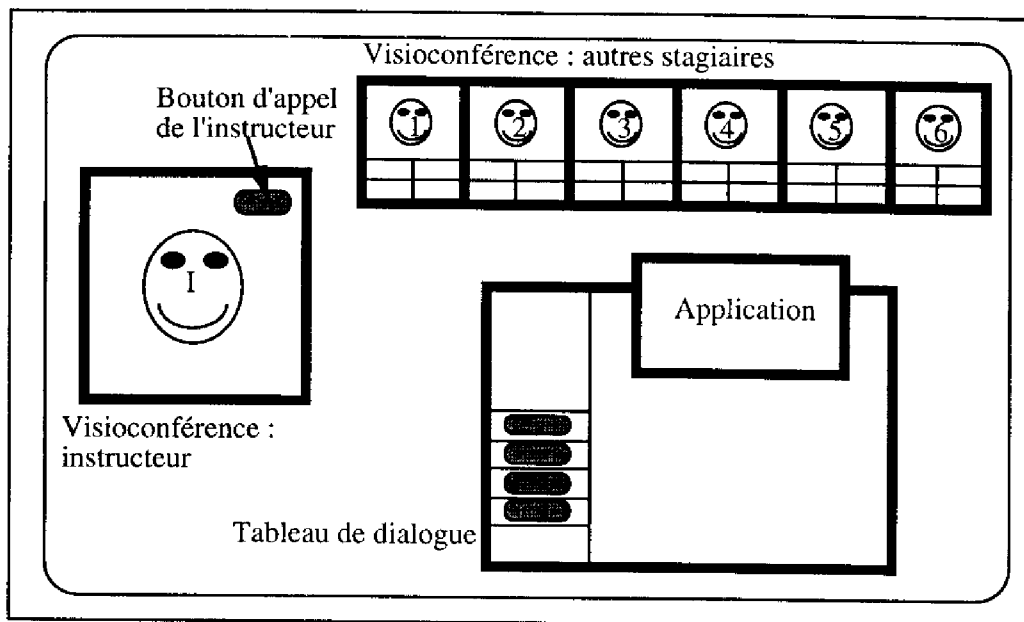


Figure V.8. Interface du poste d'un stagiaire

6. Etat des outils réalisés

L'environnement de téléformation synchrone qui vient d'être présenté dans l'étude précédente est en cours d'implémentation. En fait, étant données les difficultés techniques associées à cette réalisation, les problèmes d'implémentation sont nombreux, et à l'heure actuelle seule une maquette de l'environnement existe.

Cette maquette a été réalisée (et continue à être développée) sur une plate-forme composée de stations Sun équipées d'une carte audio Sun et d'une carte vidéo Parallax. La carte audio Sun permet la capture et la restitution de signaux sonores au format MIC (μ -law). La carte Parallax permet d'une part la capture et la restitution d'images en provenance d'une source vidéo, et d'autre part réalise la compression et la décompression temps réel de ces images au format M-JPEG. Cette carte vidéo dispose, de plus, de deux entrées vidéo ce qui autorise l'utilisation simultanée de deux sources, i.e. une caméra braquée sur l'utilisateur pour la visioconférence, et une caméra de table servant d'entrée au tableau de dialogue. Chaque station est également équipée des éléments terminaux de capture et de restitution des signaux sonores à savoir des microphones et des hauts-parleurs. Sur le plan logiciel, ces stations utilisent le système UNIX Solaris 2.x. Enfin, pour pouvoir communiquer, ces stations sont interconnectées par des réseaux hauts débits : un réseau FDDI local sur lequel a été faite la mise au point des outils, et depuis un an, un réseau ATM qui nous permet des essais sur des

réseaux longue distance. Au niveau des protocoles, la pile UDP/IP est utilisée³ car elle est à l'heure actuelle la seule qui puisse supporter les débits qui sont requis, et qui soit largement diffusée et disponible. Cette plate-forme est représentée sur la figure V.9.

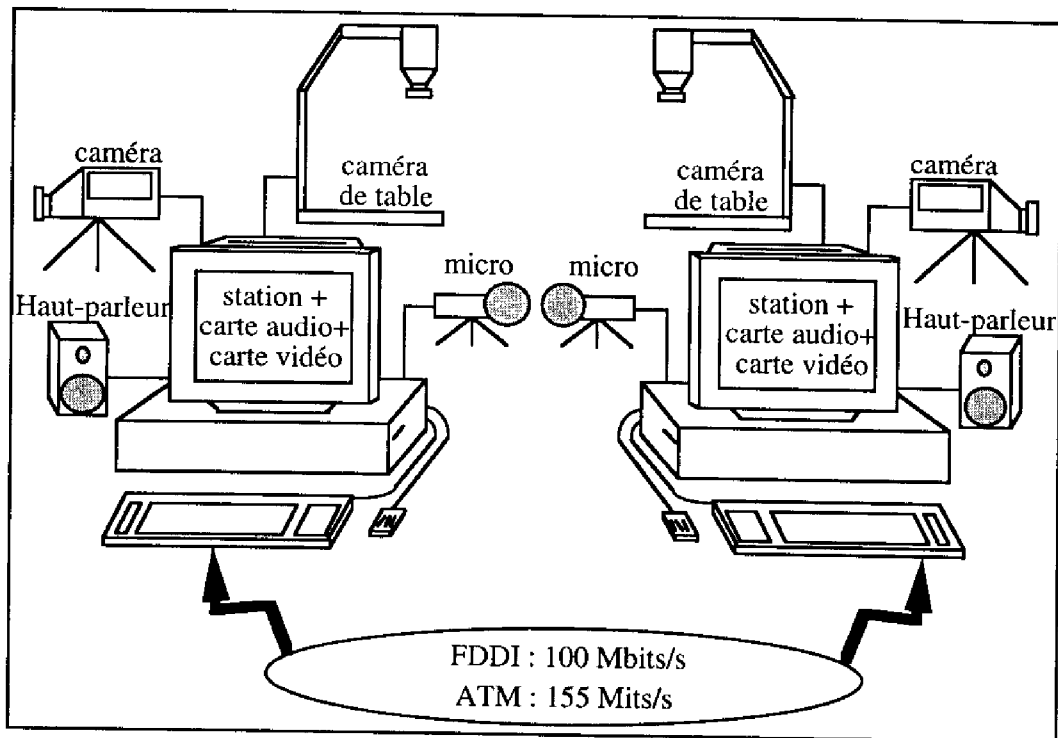


Figure V.9. Plate-forme matérielle pour l'environnement de téléformation

L'outil de visioconférence et le tableau de dialogue ont été implémentés sur cette plate-forme. Deux outils de visioconférence point à point [OWE95a] [OWE95c] [OWE96c] et un outil de visioconférence multipoints [OWE96a] [OWE96b] [OWE96d] ont été réalisés. Ils permettent le dialogue audio/vidéo entre l'instructeur et les stagiaires. La partie audio possède une qualité acceptable, équivalente à celle du téléphone numérique ; la qualité vidéo est paramétrable (en fonction des ressources disponibles) et peut s'étager de 1 à 25 images par seconde. De plus, et c'est là toute l'originalité de cet outil, les flux audio et vidéo sont synchronisés ce qui apporte une plus grande ergonomie [BLA96] et un plus grand confort d'utilisation, notamment par rapport à d'autres outils de visioconférence qui n'intègrent pas cette synchronisation multimédia.

Pour le tableau de dialogue, les fonctionnalités point à point et multipoints ont été implémentées avec toutes les fonctions associées aux télépointeurs décrites dans ce qui précède. De même, les mécanismes de synchronisation par estampilles, moteur TSPN et blocages ont été mis en place.

Il est donc dès à présent possible d'utiliser cet environnement dans le cadre de briefings ou d'auto-apprentissage. La figure V.10 représente d'ailleurs le mode de fonctionnement intégré de ces outils dans le cadre d'une interaction entre l'instructeur et un stagiaire pendant une séance d'auto-apprentissage.

³Un protocole transport à ordre partiel a aussi été utilisé [OWE95c] [OWE96c] [CHA95d].

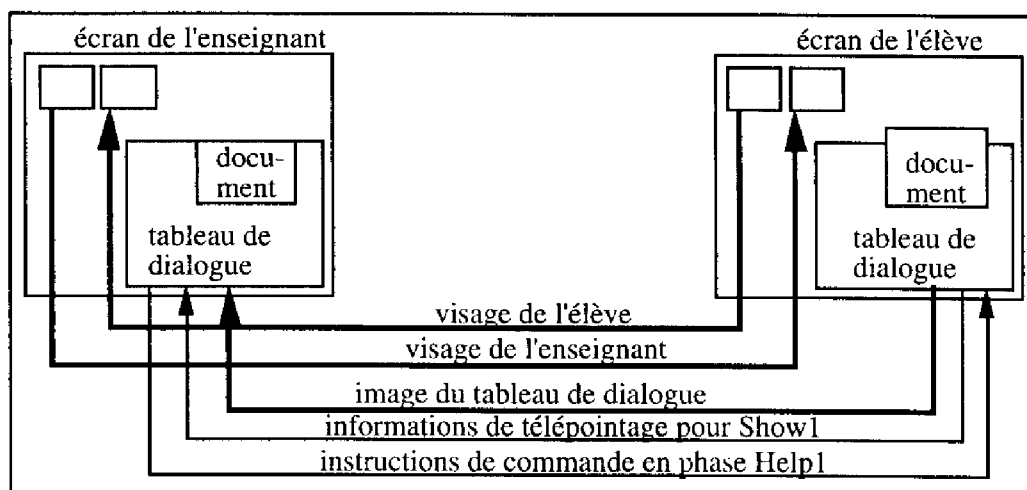


Figure V.10. Principe de fonctionnement de l'environnement de téléformation en phase d'aide spécialisée

Difficultés techniques :

Lors de la réalisation de ces outils, de très nombreuses difficultés techniques sont apparues.

- Les premières difficultés sont liées aux besoins en termes de synchronisation multimédia. En effet, il fallait synchroniser des médias continus avec des granularités très fines, et des contraintes temporelles assez strictes. Or, avec les processus Unix classiques, les overheads induits par la manipulation des contextes lors des commutations de processus rendaient impossible cette granularité de synchronisation. Aussi, il a été nécessaire d'utiliser les threads (processus allégés) [DÉM94][ANS95] [KLE96] du système Solaris 2.x pour ne plus avoir à manipuler ces contextes, et pouvoir implémenter les mécanismes de synchronisation, autant dans l'application de visioconférence que dans le tableau de dialogue [DIA94a] [OWE95c].
- Il a été également difficile d'utiliser la librairie X et la carte vidéo Parallax. En fait, cette carte et la librairie X cohabitent relativement bien (grâce aux extensions de la Xlib écrites pour la carte Parallax), mais elles sont très délicates à utiliser dans un environnement "multi-threads". En effet, la librairie X et la carte Parallax ont été conçues pour fonctionner sur un système Unix classique avec des mécanismes multitâches traditionnels. De ce fait, les primitives mises à disposition du programmeur ne sont pas ré-entrant et ne sont pas interruptibles, ce qui s'oppose totalement aux principes des mécanismes de synchronisation multimédia et du "multithreading". De ce fait, il a fallu prévenir tous les accès concurrents aux primitives de ces bibliothèques en les traitant en exclusion mutuelle, ce qui pénalise les performances des outils.
- De même, la carte audio a montré un comportement assez perturbant, avec des temps de latence substantiels, et un fonctionnement par blocs des données du flux sonore produit, ce qui accroît encore un peu le temps de latence, et nuit à l'interactivité entre les utilisateurs.
- Enfin, le support réseau utilisé (UDP/IP/Ethernet, FDDI ou ATM) n'offre aucun service temps réel pour le transport d'informations multimédias, et cette tâche est donc réalisée par l'application elle-même, ce qui la pénalise dans ses performances et dans sa simplicité d'écriture ; ces problèmes seront discutés en conclusion de cette thèse.

Conclusion

La téléformation semble constituer un enjeu de grande importance pour les années à venir. Quels que soient les domaines considérés, la technique et donc la technicité des

personnels doit toujours évoluer. Aussi, pour permettre à chacun de suivre les évolutions de son domaine, la téléformation semble être une solution très adaptée. L'étude faite au début de ce chapitre montre, par l'existence d'un nombre important de systèmes de téléformation, l'importance de ce domaine. La contribution apportée par ce chapitre se situe dans le domaine des outils à mettre en oeuvre pour rendre plus convivial et efficace ce type de système.

En fait, les fonctionnalités de base que doivent offrir les outils logiciels nécessaires à la mise en oeuvre d'une téléformation multimédia distribuée ont été extraites de l'observation des interactions instructeur/stagiaires dans un contexte de cours Airbus.

Ces outils (visioconférence et tableau de dialogue) ont été développés dans un souci de genericité. En effet, bien qu'ils aient été étudiés dans le cadre de la téléformation, ils semblent pouvoir être utilisés dans d'autres domaines : dans tous les cas pour lesquels l'intervention de plusieurs personnes sur un support informatique est nécessaire. Il peut s'agir, par exemple :

- de la conception de documents par un groupe de personnes, comme par exemple :
 - une maquette publicitaire dont la réalisation fait intervenir le concepteur, un ou plusieurs graphistes et le commanditaire de la maquette ;
 - des modules de cours dont la définition nécessite l'intervention de plusieurs spécialistes ;
 - des schémas de conception de systèmes électriques, électroniques, ... faisant intervenir plusieurs concepteurs. Les projets européens ont pour but de faire travailler ensemble différentes sociétés, et des déplacements pourraient être évités par l'utilisation de ces outils.
- du pilotage/surveillance d'unités de production ou d'expériences embarquées du type de celles qui sont réalisées dans une station orbitale. Plusieurs experts peuvent intervenir sur ces unités, ceux-ci n'étant pas géographiquement au même endroit ;
- de systèmes de diagnostics médicaux faisant intervenir des spécialistes distants ;
- etc. ...

Toutefois, pour pouvoir s'adapter à d'autres domaines de téléformation ou au travail coopératif des études doivent être menées pour savoir si les fonctionnalités données ici s'adaptent bien, et si d'autres fonctionnalités ne doivent pas être ajoutées à celles-ci. De plus, ces outils devront être validés dans le cadre réel des cours pour savoir si les interactions proposées répondent aux exigences réelles de la téléformation aéronautique ; ce travail sera mené dans le cadre du projet TOPASE.



Conclusion générale

Au cours de cette thèse, l'objectif a été de concevoir de façon approfondie et de développer quelques outils pour la mise en oeuvre de systèmes distribués multimédias coopératifs. L'application de base qui a illustré les concepts analysés et les travaux réalisés est la visioconférence. La problématique que nous nous sommes fixée consistait à concevoir des mécanismes de synchronisation multimédia, de gestion de la QoS et de coopération ; les mécanismes que nous avons développés ont été appliqués et ont donné naissance à plusieurs prototypes de systèmes de visioconférence : TSVS, PNSVS 1, PNSVS 2 et N-TSVS.

Contributions de cette thèse

L'étude de la double problématique associée à la manipulation de données multimédias et à l'aspect coopératif (chapitre 1) a clairement établi les contraintes et les difficultés inhérentes à la conception de ce type d'applications. De plus, en voulant réaliser cette étude dans le cadre des systèmes distribués asynchrones - qui sont les systèmes les plus répandus et qui représentent l'étude la plus générale possible - nous avons été du même coup amenés à ne considérer que des hypothèses minimales ; ce travail s'en est trouvé d'autant plus complexe puisque les systèmes sur lesquels nous travaillons ne présentent aucune propriété de référence, le propre d'un système asynchrone étant de ne garantir aucun service. A cause de cette complexité, un modèle formel s'est avéré nécessaire pour exprimer les contraintes de synchronisation multimédia ; ce modèle a été choisi pour son pouvoir de modélisation et son pouvoir d'expression : il devait être capable de modéliser de façon simple et complète les contraintes de synchronisation qui existent dans la présentation des flux d'une application multimédia. Le choix s'est porté sur le modèle TSPN qui est simple, qui a le pouvoir d'expression escompté, et surtout le meilleur pouvoir de modélisation de tous les modèles étudiés. En ce qui concerne l'activité coopérative, un modèle est nécessaire pour représenter les interactions possibles entre les différents participants d'un groupe coopératif, les différentes sous-tâches ou sous-activités, leurs participants et les relations entre ces participants et ces sous-activités. Le modèle retenu est un graphe coopératif dont la sémantique est basée sur le partage de données, et qui prend en compte à la fois des domaines, des sous-domaines et des groupes de coopération.

En partant de cette étude, les principales applications des principaux domaines d'application ont été confrontées à la double problématique posée, et ont montré leurs lacunes conceptuelles pour résoudre les problèmes inhérents aux applications multimédias coopératives, autant au niveau de la manipulation et du traitement des données multimédias qu'au niveau de la gestion de l'activité coopérative.

Aussi, les travaux présentés dans le chapitre 2 ont résolu le problème de la synchronisation multimédia en environnement asynchrone, et ont dû, pour cela, tenir

compte des nombreux indéterminismes au niveau du support de communication, du système opératoire et des cartes de présentation multimédia. Les mécanismes avancés des nouveaux systèmes opératoires ont été utilisés (et sont indispensables pour résoudre ce genre de problèmes), comme les threads, la classe d'ordonnancement temps réel et les outils de synchronisation des processus (mutex, variables conditions, etc.).

De même, des mécanismes ont été développés pour optimiser la qualité de présentation des flux de l'application. Pour cela, une gestion des tampons optimale a tout d'abord été réalisée ; toutefois, même si son apport est indéniable, des perturbations peuvent dégrader son efficacité, et une nouvelle architecture de synchronisation a été développée. Cette architecture est basée sur un transport à ordre partiel dont les mécanismes de fiabilité partielle, de perte et de délivrance au plus tôt permettent d'éviter toute dégradation abusive.

Pour le dernier paramètre de QoS - le contrôle du retard de présentation de bout en bout - des mécanismes ont été proposés. Ils permettent un contrôle "best effort" de ce paramètre, mais ne peuvent en aucun cas le garantir. Avec les hypothèses minimalistes qui ont été prises, il semble impossible de pouvoir garantir un tel paramètre de QoS.

Toutefois, tous ces mécanismes de gestion de la QoS, de la synchronisation multimédia et du retard de bout en bout impliquent un comportement très particulier des processus de stockage, de présentation et d'orchestration. Pour modéliser ce comportement, le modèle TSPN a été dérivé en plusieurs TSPN applicatifs, TSPN de pré-synchronisation et réseau de Petri des ordres de délivrance du transport, de façon à modéliser finement les comportements de chacun des processus de chacune des couches de cette architecture ; la différence entre chacun de ces TSPN est d'ailleurs assez frappante ; ce résultat, qui est précurseur dans le domaine, montre combien le comportement asynchrone des systèmes distribués oblige à des ordonnancements de tâches différents, adaptés et subtils entre les couches et sous-couches.

Le chapitre 4 s'est intéressé, quant à lui, à la présentation des mécanismes et de l'architecture d'une application de visioconférence coopérative (N-TSVS). Cette application se décompose en deux parties : une partie de gestion de l'activité coopérative (gestion des groupes et de leur dynamique, des participants, des relations et des interactions qu'ils peuvent avoir entre eux, etc.) et une partie propre à la visioconférence avec les problèmes classiques des applications multimédias de gestion de la qualité de service, de la synchronisation, et les problèmes de communications de groupes multimédias.

Enfin, le chapitre 5 met en oeuvre les propos tenus dans les premiers chapitres dans une application plus complexe que la visioconférence : il s'agit d'un environnement de téléformation aéronautique synchrone. En fait, le développement de cette application part d'une potentialité de modifier le système traditionnel de formation des pilotes et agents de maintenance sur la gamme des avions Airbus. Pour cela, nous avons été amenés à prendre en compte le comportement des étudiants et des instructeurs, à étudier les relations et les interactions entre les participants à un cours, pour essayer de les reproduire au travers de stations de travail et de réseaux informatiques. Des interfaces pour les postes instructeurs et élèves ont été définies de façon à rendre facilement accessibles les différentes fonctionnalités offertes par ce système. De plus, ce système a été partiellement réalisé et les détails d'implémentation présentés.

Bilan général

Le travail de conception d'applications multimédias coopératives réalisé dans cette thèse représente à la fois un travail de développement d'études et un manuel d'intégration des recherches menées dans CESAME. En effet, dans les travaux de cette thèse ont été intégrés les travaux menés dans la tâche synchronisation (avec l'utilisation du modèle TSPN), les travaux effectués dans la tâche coopération (avec l'utilisation du modèle à base de graphes coopératifs et avec l'utilisation des modules de gestion des entrées et sorties dynamiques et de contrôle de l'activité coopérative) et les travaux réalisés dans la tâche transport haute vitesse (avec l'utilisation d'un transport à ordre

partiel). Toutefois, la conception d'applications distribuées multimédias coopératives a fortement influencé les tâches synchronisation, coopération et transport puisque les concepts développés dans ces tâches découlent des contraintes et des besoins de ces mêmes applications.

Ce travail s'est donc avéré relativement large, et a nécessité plusieurs domaines de compétences ; il a réclamé à la fois des connaissances en réseaux de Petri lors de toutes les phases de modélisation, et des connaissances informatiques lors de la conception de tous les mécanismes développés dans cette thèse, puisqu'ils ont toujours été confrontés à la problématique posée et aux caractéristiques des systèmes utilisés. D'un point de vue purement mise en oeuvre des implémentations ont été réalisées sur la plate-forme du LAAS et ont nécessité des connaissances pluri-disciplinaires : elles ont nécessité de connaître parfaitement les mécanismes de multi-threading, d'ordonnancement temps-réel, les streams, les outils de synchronisation avancés, les mécanismes de communication inter-processus (locaux ou distants), les systèmes de multi-fenêtrage, les algorithmes de compression vidéo, la manipulation de l'audio et de la vidéo, le réseau de communication ATM (avec IP sur ATM ou ATM natif directement)...

Perspectives

Les travaux exposés dans cette thèse ont résolu plusieurs problèmes posés par la conception d'applications multimédias coopératives. Au niveau de la synchronisation multimédia - à une hypothèse sur la constance des temps de latence des cartes de présentation multimédias près - les mécanismes développés garantissent le respect de toutes les contraintes de synchronisation intra et inter-flux. Au niveau de la QoS, il en est de même, à l'exception du retard de présentation de bout en bout ; les mécanismes proposés ne garantissent pas le respect de ce paramètre de QoS et laissent donc ce problème ouvert : peut-on sur une plate-forme asynchrone, garantir le respect de cette contrainte, et si oui, quels sont les mécanismes qui permettraient d'y parvenir ?

Au niveau de la coopération, les travaux décrits ne traitent que des problèmes de gestion des groupes coopératifs et de communication dans ces groupes d'un point de vue informatique et télécommunications. Par contre, l'activité coopérative ne relève pas que de ces domaines, mais relève également des sciences sociales et cognitives. Ainsi, même si les solutions que nous avons proposées sont intéressantes, elles tireraient un grand profit d'une étude sociale et cognitive qui permettrait de les rendre plus adaptées et plus efficacement utilisables par des utilisateurs non informaticiens.

De même, l'implémentation de N-PNSVS - une application de visioconférence coopérative synchronisée en utilisant les mécanismes modélisés par les TSPN du chapitre II - n'a pas été réalisée ; ce travail s'est arrêté à la phase de conception. Cette implémentation reste malgré tout une des perspectives intéressantes de ces travaux.

Au niveau des mécanismes du transport à ordre partiel utilisés, l'implémentation de PNSVS 2 qui a été réalisée n'utilise qu'un prototype d'implémentation ; de ce fait, elle bénéficierait certainement de l'efficacité et des performances d'une implémentation industrielle. De plus, tous les mécanismes de gestion de l'ordre et de la fiabilité partielle n'ont pas été étudiés, ni implantés ; une étude visant à promouvoir d'autres types de gestion reste donc ouverte afin de voir si la visioconférence ne tirerait pas plus d'avantages de ces autres formes de gestion. De plus, les travaux quant à un transport à ordre partiel multicast ne sont pas suffisamment avancés. Il serait toutefois très intéressant de voir si dans une application de visioconférence multipoints, un tel transport aurait la même utilité que celle qu'il a en point à point. Même si dans la partie 4 nous mettons cette hypothèse en avant, la confirmation ne pourra se faire qu'avec la réalisation effective de ce logiciel, ou par une évaluation très fine.

Enfin, au niveau de l'environnement de télé-formation synchrone, l'implémentation du prototype est en cours. L'étude qui a précédé le début de cette réalisation n'est toutefois pas achevée et se poursuit encore aujourd'hui, notamment en fonction des contraintes imposées par la plate-forme de développement, mais surtout en fonction des

exigences et des comportements observés des instructeurs et des stagiaires. La finalisation de cet environnement est en cours. De plus, une expérimentation réelle de ce système devra être faite afin de le valider, puis éventuellement de l'étendre. En fait, ce travail est prévu dans le cadre du projet TOPASE.

Bibliographie de l'auteur

Rapport de DEA

- [OWE93] P. Owezarski, "Etude et Spécification de Systèmes de Communications de Groupes pour les Applications Coopératives", rapport de stage de DEA Interface Homme-Système MultiModale (IHS2M), Université Paul Sabatier, Juin 1993

Contributions à ouvrages

- [OWE95a] P. Owezarski, M. Diaz, P. Sénac, "Modélisation et implémentation de mécanismes de synchronisation multimédia dans une application de visioconférence", Actes du colloque francophone sur l'ingénierie des protocoles (CFIP'95), pp 305-319, éditions Hermès, Rennes, France, 10-12 mai 1995
- [OWE96c] P. Owezarski, M. Diaz, "Models for enforcing multimedia synchronization 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

Conférences internationales

- [OWE95b] P. Owezarski, V. Baudin, M. Diaz, J.F. Schmidt, "Multimédia Teleteaching : Introduction of Synchronization and Cooperation Mechanisms in Distance Learning", Proceedings of the world conference on educational multimedia and hypermedia (ED'MEDIA 95), pp 517-522, Graz, Austria, 17-21, 1995
- [BAU96b] V. Baudin, M. Diaz, P. Owezarski, T. Villemur, "Design and realization of a synchronous cooperative shared electronic board", proceedings of the Advanced Technology Workshop (ATW'96), Toulouse, France, July 8-10, 1996

Conférences francophones

- [BAU95] V. Baudin, P. Owezarski, M. Diaz, "Projet CESAME : conception formelle de systèmes multimédias coopératifs à hauts débits", Actes des

journées réseaux (JRES'95), pp 37-44, Chambéry, France, 22-24 novembre 1995

- [OWE96a] P. Owezarski, T. Villemur, M. Diaz, "Conception d'un système de visioconférence coopératif", Actes des 8ème rencontres francophones sur le parallélisme (RenPar'8), pages 25-28, Bordeaux, France, 20-24 Mai 1996
- [OWE96b] P. Owezarski, T. Villemur, M. Diaz, "Conception et implémentation d'un système de visioconférence coopératif à N intervenants", Actes des journées de recherche sur le contrôle réparti dans les applications coopératives (CRAC'96), pages 87-92, Paris, France, 30-31 Mai 1996

Séminaires francophones

- [OWE96g] P. Owezarski, V. Baudin, M. Diaz, "Conception et développement d'un système synchrone de téléformation professionnelle", Actes des séminaires hypermédia, éducation et formation, Paris, France, 22 mars 1996

Rapports de contrat

- [DIA94b] M. Diaz, P. Owezarski, T. Villemur, "Une Définition Logique de la Coopération Basée sur le Partage des Données", Projet CESAME. Rapport du marché CNET France Télécom 92 1B 178 - Lot 3. Rapport interne LAAS no. 94008, 19 pages, Janvier 1994

Rapports LAAS

- [BAU96a] V. Baudin, M. Diaz, P. Owezarski, "Extensions proposal for IconAuthor", Rapport LAAS N° 96288, 6 pages, Juillet 1996
- [DIA94a] M. Diaz, P. Owezarski, "Développement d'un système de visioconférence sur réseau local", Rapport LAAS No 94354, 33 pages, Juillet 1994
- [OWE95c] P. Owezarski, M. Diaz, "Gestion de la synchronisation multimédia et de la qualité de service dans une application de visioconférence : mécanismes système et transport", Rapport LAAS No. 95460, 62 pages, Novembre 1995
- [OWE96d] P. Owezarski, T. Villemur, M. Diaz, "Conception et implémentation d'un système de visioconférence coopératif", Rapport LAAS N° 96401, Août 1996
- [OWE96e] P. Owezarski, M. Diaz, "Models for enforcing multimedia synchronization and QoS in visioconference applications", LAAS report No 96287, 16 pages, July 1996
- [OWE96f] P. Owezarski, V. Baudin, M. Diaz, "Conception et réalisation d'un environnement de téléformation synchrone", Rapport LAAS No 96285, 27 pages, Juillet 1996

Bibliographie

- [AAL94] M. P. van de Aalst, K. M. van Hee, G. J. Houben, "Modelling Workflow Management with High Level Petri Nets", Workshop on Computer Supported Cooperative Work, Petri Nets and related formalisms, Saragoza, pages 31-50, June 1994
- [ADE94] B. Adelberg, H. Garcia-Molina, B. Kao, "Emulating Soft Real Time Scheduling Using Traditional Operating System Schedulers", Proceedings of the Real Time Systems Symposium, San Juan, Puerto Rico, December 7-9, 1994
- [AGO94] A. Agostini, G. de Michelis, K. Petruni, "Keeping Workflow Models as Simple as Possible", Workshop on Computer Supported Cooperative Work, Petri Nets and related formalisms, Saragoza, pages 11-29, June 1994
- [AHA91] M. Ahamad, M. H. Ammar, S. Y. Cheung, "Multidimensionnal Voting", ACM Transactions on Computers Systems, Vol. 9, No. 4, pp 399-431, November 1991
- [ALL83] J.F. Allen, "Maintaining knowledge about temporal intervals", Communications of the ACM, Vol. 26, No. 11, November 1983
- [AME93a] P.D. Amer, C. Chassot, T. Connolly, M. Diaz, "Partial order transport service for multimedia applications : reliable service", proceedings of the second High Performance Distributed Computing Conference, July 1993
- [AME93b] P.D. Amer, C. Chassot, T. Connolly, M. Diaz, "Partial order transport service for multimedia applications : unreliable service", proceedings of the 3rd International Networking Conference, INET'93, August 1993
- [AME94a] P.D. Amer, C. Chassot, T. Connolly, P. Conrad, M. Diaz, "Partial order transport service for multimedia and other applications", IEEE/ACM transactions on Networking, Vol. 2, No. 5, October 1994
- [AME94b] P. Amer, C. Chassot, T. Connolly, M. Diaz, A. Lozes, "Service d'ordre partiel non fiable" Rapport LAAS No. 94028, 1994
- [AMI93] A. Aminmansour, "Multimedia : A revolutionary tool to enhance teaching and learning of structural steel design", Proceedings of ED'MEDIA 93, 1993
- [AND95] K. Andrews, "Embedding Courseware into the Internet : Problems and Solutions", Proceedings of ED'MEDIA 95, 1995
- [ANS95] Sun Microsystems, "SunOS5.4 : Guide to Multithread Programming", 1995

- [ATE96] A.F. Ates, M. Bilgic, S. Saito, B. Sarikaya, "Using timed CSP for specification, verification and simulation of multimedia synchronization", IEEE journal on selected areas in communications, Vol. 14, No. 1, January 1996
- [AUG93] F. Augenstein, T. Ottman, J. Schoning, "Logical markup for hypermedia documents : the TRAIN system", proceedings of ED'MEDIA 93, 1993
- [BAK94] T.P. Baker, F. Mueller, V. Rustagi, 'Experience with a Prototype of the POSIX Minimal Realtime System Profile", IEEE Newsletter on Real Time Systems, 1994
- [BAN89] L. J. Bannon, K. Schmidt, "CSCW: Four Characters in Search of a Context", Studies in Computer Supported Cooperative Work: Theory, Practice and Design, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991
- [BAR93a] P. Barker, S. Giller, S. Richards, A. Banerji, C. Emery, "Interactive technologies for language learning", proceedings of ED'MEDIA 93, 1993
- [BAR93b] A.E. Barron, K. Ivers, "Simulation training for electronic mail systems", proceedings of ED'MEDIA 93, 1993
- [BAT94] A.W. Bates, "Educational Multi-Media in a Networked Society", Proceedings of ED'MEDIA 94, 1994
- [BAU95] V. Baudin, P. Owezarski, M. Diaz, "Projet CESAME : conception formelle de systèmes multimédias coopératifs à hauts débits", Actes des journées réseaux (JRES'95), pp 37-44, Chambéry, France, 22-24 novembre 1995
- [BAU96a] V. Baudin, M. Diaz, P. Owezarski, "Extensions proposal for IconAuthor", Rapport LAAS N° 96288, Juillet 1996
- [BAU96b] V. Baudin, M. Diaz, P. Owezarski, T. Villemur, "Design and realization of a synchronous cooperative shared electronic board", proceedings of the Advanced Technology Workshop (ATW'96), Toulouse, France, July 8-10, 1996
- [BEL94] T. Beltran, A. Péninou, "Hypermedia Tutoring System : Towards an Architecture Dedicated to the WEB Specification", Proceedings of ED'MEDIA 94, 1994
- [BEN92] S. Benford, H. Smith, A. Shepherd, A. Bullock, H. Howidi, "Information Sharing Approach to CSCW: The Grace Project", Computer Communications, Vol. 15, No. 8, pp 502-508, October 1992
- [BEN93] S. Benford, J. Palme, "A Standard for OSI Group Communication", Computer Networks and ISDN systems, Vol. 25, No. 8, pp 933-946, Mars 1993
- [BER91] B. Berthomieu, M. Diaz, "Modelling and verification of time dependant systems using time Petri nets", IEEE transactions on software engineering, Vol. 17, No. 3, March 1991
- [BER93] Berkom, "The BERKOM multimedia mail teleservice", Technical report, May 1993
- [BIG89] C. Bignoli, C. Simone, "AI Techniques for Supporting Human to Human Communications in CHAOS", Studies in Computer Supported Cooperative Work: Theory, Practice and Design, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991
- [BIR87] K. P. Birman, T. A. Joseph, "Reliable Communication in the Presence of Failures", ACM Transactions on Computers Systems, Vol. 5, No. 1, pp 47-76, February 1987

- [BIR91] K. Birman, A. Schiper, P. Stephenson, "Lightweight Causal and Atomic Group Multicast", *ACM Transactions on Computers Systems*, Vol. 9, No. 3, pp 272-314, August 1991
- [BLA92] G. Blakowski, J. Hübel, U. Langrehr, M. Mülhäuser, "Tool support for the synchronisation and presentation of distributed multimedia", *Computer Communication*, Vol. 15, pp 611-618, December 1992
- [BLA94] G. S. Blair, T. Rodden, "The Opportunities and Challenges of CSCW", *Journal of the Brazilian Computer Society*, Vol. 1, No. 1, pp 3-14, Juillet 1994
- [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
- [BOR92] N. Borenstein, N. Freed, "MIME (Multipurpose Internet Mail Extensions) : Mechanisms for specifying and describing the format of Internet message bodies", Request For Comment No. 1341, June 1992
- [BOY96a] M. Boyer, "Renégociation de qualité de service dans une application de visioconférence : PNSVS", Mémoire d'ingénieur de l'Ecole Nationale Supérieure d'Electronique, d'Electrotechnique, d'Informatique et d'Hydraulique de Toulouse (ENSEEIH), Juin 1996
- [BOY96b] M. Boyer, "Analyse de la synchronisation de l'application de visioconférence avec renégociation dynamique de QoS : PNSVS 2", Rapport de stage de DEA de l'Institut National Polytechnique de Toulouse (INPT), Septembre 1996
- [CCI88] CCITT, "Terms and definition related to the quality of telecommunication services", CCITT E.800, Blue book, 1988
- [CHA95a] C. Chassot, M. Diaz, A. Lozes, "From the partial order concept to partial order multimedia connections", *Journal of high speed network (JHSN)*, 1995
- [CHA95b] C. Chassot, M. Diaz, A. Lozes, "Principes d'implantation d'une connexion multimédia d'ordre partiel", 4ème colloque francophone sur l'ingénierie des protocoles, CFIP'95, Rennes, France, 10-12 mai 1995
- [CHA95c] C. Chassot, M. Fournier, M. Diaz, A. Lozes, "Service definition of multimedia partial order connections", proceedings of the COST 237 conference, Copenhagen, Denmark, November 21-22, 1995
- [CHA95d] C. Chassot, "Architecture de transport multimédia à connexions d'ordre partiel", Thèse de doctorat de l'Institut National Polytechnique de Toulouse (INPT), Décembre 1995
- [CHE85] D.R. Cheriton, W. Zwaenepoel, "Distributed Process Groups in the V Kernel", *ACM Transactions on Computers Systems*, Vol. 3, No. 2, pp 77-107, May 1985
- [CHE88] D.R. Cheriton, "The V Distributed System", *Communications of the ACM*, Vol. 31, No. 3, pp 314-333, Mars 1988
- [CHE90] S. Y. Cheung, M. Ahamad, M. H. Ammar, "Multi-Dimensional Voting: A General Method for Implementing Synchronization in Distributed Systems", *Proceedings of the IEEE: 10th International Conference on Distributed Computing Systems*, pages 362-369, May 1990
- [CHU94] G. Chung, K. Jeffay, H. Abdel-Wahab, "Dynamic participation in a computer based conferencing system", *Computer communications*, Vol. 17, No. 1, January 1994

- [CIC95] G. Cicchelerio, "Synchronisation pour une présentation multimédia", Mémoire d'ingénieur du Conservatoire National des Arts et Métiers (CNAM), 1995
- [COU94a] G. Coulson, G. S. Blair, P. Robin, "Micro-kernel support for continuous media in distributed systems", *Computer Networks and ISDN systems*, Vol. 26, 1994
- [COU94b] J.P. Courtiat, R.C. de Oliveira, L.F.R.C. Carmo, "Towards a new multimedia synchronization mechanism and its formal specification", *Proceedings of ACM multimedia'94*, San Francisco, October 1994, pp 133-140
- [COU96a] J.P. Courtiat, L.F. Rust da Costa Carmo, R. Cruz de Oliveira, "A general-purpose multimedia synchronization mechanism based on causal relations", *IEEE journal on selected areas in communications*, Vol. 14, No. 1, January 1996
- [COU96b] J.P. Courtiat, R.C. de Oliveira, "Proving temporal consistency in a new multimedia model", *Proceedings of ACM multimedia'96*, Boston, November 18-22, 1996
- [CRO94] P. Croisy, D. Clement, L. Barme, "Co-Learning at a distance : FIRST trial of an integrated learning environment", *Proceedings of ED'MEDIA 94*, 1994
- [DAG94] F. Dagorn, C. Gross, "World-Wide-Web", disponible sur l'URL <http://www.urec.fr>, 1994
- [DAI94] L. Dairaine, "Techniques de synchronisation pour les communications dans les systèmes haut-débit multimédia", thèse de doctorat de l'université Pierre et Marie Curie (Paris VI), Septembre 1994
- [DEE88] S. Deering, "Host extensions for IP multicasting", Request For Comments No. 1054, 1988
- [DEM94] I. Demeure, J. Farhat, "Systèmes de processus légers : concepts et exemples", *Techniques et sciences informatiques*, Vol. 13, No. 6, 1994
- [DER93] A.C. Derycke, P. Croisy, P. Vilers, "Computer Supported Cooperative Learning : a real time multimedia approach", *proceedings of ED'MEDIA 93*, 1993
- [DIA92] M. Diaz, "A logical model of cooperation", *Proceedings of the IEEE: Third Workshop on Future Trends of Distributed Computing Systems*, pages 64-70, April 1992
- [DIA93a] M. Diaz, P. Sénac, "Time stream Petri nets, a model for multimedia streams synchronization", *proceedings of MultiMedia modelling'93*, Singapore, November 1993
- [DIA93b] M. Diaz, P. Sénac, P. De Saqui-Sannes, "Un modèle formel pour la spécification de la synchronisation multimédia en environnement distribué", *Actes du colloque francophone sur l'ingénierie des protocoles*, Montréal, Canada, 7-9 septembre 1993
- [DIA93c] M. Diaz, T. Villemur, "Services et Protocoles d'Adhésion à des Groupes Coopératifs Dynamiques", *Projet CESAME. Rapport du marché CNET France Télécom 92 1B 178 - Lot 3. Rapport interne LAAS no. 93274*, 48 pages, Juillet 1993
- [DIA93d] M. Diaz, T. Villemur, "Membership Services and Protocols for Cooperative Frameworks of Processes", *Computer Communications*, Vol. 16, No. 9, pp 548-556, September 1993
- [DIA93e] M. Diaz, "Coopération, Logique et Partage de Données", *Actes de l'AFCEt et de l'AFIA. Premières Journées Francophones Intelligence*

- Artificielle Distribuée et Systèmes Multi-Agents, pages 253-262, Avril 1993
- [DIA94a] M. Diaz, P. Owezarski, "Développement d'un système de visioconférence sur réseau local", Rapport LAAS No. 94354, Juillet 1994
- [DIA94b] M. Diaz, P. Owezarski, T. Villemur, "Une Définition Logique de la Coopération Basée sur le Partage des Données", Projet CESAME, Rapport du marché CNET France Télécom 92 1B 178 - Lot 3, Rapport LAAS no. 94008, 19 pages, Janvier 1994
- [DIA94c] M. Diaz, A. Lozes, C. Chassot, P.D. Amer, "Partial order connections : A new concept for high speed and multimedia services and protocols", Annals of telecommunications, Vol. 49, No. 5-6, May-June, 1994
- [DIA94d] M. Diaz, G. Pays, "The CESAME project : formal design of high speed multimedia cooperative systems", Annals of telecommunications, Vol. 49, No. 5-6, may-june, 1994
- [DIA94e] 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, pages 219-238, Zaragoza, June 1994
- [DIA95] M. Diaz, K. Drira, A. Lozes, C. Chassot, "Definition and representation of the quality of service for multimedia systems", 6th International conference on high speed Networking, HPN'95, Palma de Mallorca (Balearic Islands), Spain, September 11-15, 1995
- [DOL89] J. Dollimore, S. Wilbur, "Experiences in Building a Configurable CSCW System", Studies in Computer Supported Cooperative Work: Theory, Practice and Design, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991
- [DOR95] T. Dorcey, "The CU-SeeMe desktop videoconferencing software", ConneXions - The interoperability report, Vol. 9, No. 3, pages 42-45, March 1995
- [DRU92] D.L. Drucker, M.D. Murie, "QuickTime Handbook", Hayden, 1992
- [DUR87] E. H. Durfee, V. R. Lesser, D. D. Corkill, "Coherent Cooperation Among Communicating Problem Solvers", IEEE Transactions on Computers, Vol. 36, No. 11, pp 1275-1291, Novembre 1987
- [DUR89] E. H. Durfee, V. R. Lesser, D. D. Corkill, "Trends in Cooperative Distributed Problem Solving", IEEE Transactions on Knowledge and Data Engineering, Vol. 1, No. 1, pp 63-83, Mars 1989
- [ELD94] J. Eldred, " Distance Management Education : Process and Evaluation of Course Design, Manager Support and Media", Proceedings of ED'MEDIA 94, 1994
- [ELL91] C. Ellis, S. Gibbs, G. Rein, "Groupware. Some issues and experiences", Communications of the ACM, Vol. 34, No. 1, pp 38-58, Janvier 1991
- [ELL94] A. Ellis, R.S. Debrecey, "Electronic Classroom : Features, Users and Evaluation Studies", Proceedings of ED'MEDIA 94, 1994
- [ESC95] G. Eschelbeck, "An architecture for Multimedia Communication in a Distributed Education Environment", Proceedings of ED'MEDIA 95, 1995
- [EYK92] J.R. Eykholt, S.R. Kleiman, S. Barton, R. Faulkner, A. Shivalingiah, M. Smith, D. Stein, J. Voll, M. Weeks, D. Williams, "Beyond Multiprocessing... Multithreading the SunOS Kernel, Summer'92 USENIX, San Antonio, TX, June 8-12, 1992

- [FAN96] C. fantini, "Etude et réalisation d'un prototype joueur de scénarii multimédias", Mémoire d'ingénieur du Conservatoire National des Arts et Métiers (CNAM) en automatique industrielle, 1996
- [FIL93] M. Filali, "Une approche formelle pour la description du comportement temporel des applications multimédia", actes du colloque francophone sur l'ingénierie des protocoles (CFIP'93), Montréal, Canada, 7-9 septembre 1993
- [FIL94] M. Filali, "The specification of the temporal behaviour for multimedia scheduling", annales des télécommunications, tome 49, No. 5-6, mai-juin 1994
- [FRE94] R. Frederick, "Experiences with real-time software video compression", 6th International workshop on packet Video, Portland, Oregon, September 26-27, 1994
- [FUK94] H. Fuks, L. Mendonça de Moura, "A Document Based Approach for Cooperation", Journal of the Brazilian Computer Society, Vol. 1, No. 1, pp 36-45, July 1994
- [G711] CCITT, "Recommandation G711 : Codage de signaux analogiques en modulation par impulsions et codage", Rec. G.711, Fascicule III.4, Genève, 1972
- [GRE94] J.M. Greenberg, "Integrated Multimedia in Distance Education", Proceedings of ED'MEDIA 94, 1994
- [GUI92] Y. Guillot, J. Guivarc'h, T. Morin, J. Seguin, "Télé-Amphi : un service de communication de groupe", L'écho des RECHERCHES, Vol. 149, pp 3-12, 3ème trimestre 1992
- [HAH89] U. Hahn, M. Jarke, S. Eherer, K. Kreplin, "CoAUTHOR - A Hypermedia Group Authoring Environment", Studies in Computer Supported Cooperative Work: Theory, Practice and Design, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991
- [HAM96] R. Hamon, "Interface graphique pour visioconférence synchronisée à N utilisateurs", Rapport de stage ingénieur de l'Ecole Nationale de l'Aviation Civile (ENAC), Juin 1996
- [HAN94] V.L. Hanson, "Internetworking for K-12 Education", Proceedings of ED'MEDIA 94, 1994
- [HEN89] P. Hennessy, "Information Domains in CSCW", Studies in Computer Supported Cooperative Work: Theory, Practice and Design, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991
- [HEW91] C. Hewitt, J. Inman, "DAI Betwixt and Between: From "Intelligent Agents" to Open System Science" IEEE Transactions on Systems, Man, and Cybernetics, Vol. 31, No. 6, pp 1409-1419, November/Décember 1991
- [HGT92] Hock-Guan Tan, "ISDN Interactive Distance Learning - A Step Closer To Near Paperless Learning Environment ?", proceedings of Information processing 92, volume 2 : Education and society, 1992
- [HOD89] M.E. Hodges, R.M. Sasnett, M.S. Ackerman, "Athena Muse: a construction set for multi-media applications", IEEE software, pp 37-43, January 1989
- [HOS93] N. Hoskin, R.M. Aiken "CO-EXPLAINER : A Machine-Learning Companion" proceedings of Information processing 92, volume 2 : Education and society, 1993

- [HUA96] C.M. Huang, C.M. Lo, "An EFSM- based multimedia synchronization model and the authoring system", IEEE journal on selected areas in communications, Vol. 14, No. 1, January 1996
- [HUG89] L. Hughes, "Multicast Response Handling Taxonomy", Computer Communications, Vol. 12, No. 1, pp 39-46, February 1989
- [INT91] D. J. Mac Intosh, S. E. Conry, R. A Meyer, "Distributed Automated Reasoning: Issues in Coordination, Cooperation, and Performances", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 6, pp 1307-1316, November/Décember 1991
- [ISO2031] ISO/TC97/SC21 WG1 N2031 (1987), "Working Draft Addendum to ISO 7498-1 on Multipeer Data Transmission", 1987
- [ISO90] ISO-IEC JTC1/SC2/WG12, "Coded representation of multimedia and hypermedia information", Multimedia and hypermedia information coding expert group (MHEG), working document version 2, July 1990
- [ISO9074] ISO/IEC ISO 9074: 1989 (E). Information processing systems. Open System Interconnection. Estelle: A formal description technique based on an extended state transition model
- [ISO92] ISO/IEC, "Hypermedia/time-based document structuring language (HyTime)", International Standard ISO/IEC IS10744, 1992
- [ISO93] ISO, "Coded representation of multimedia and hypermedia information objects (MHEG)", Committee draft, ISO/IEC CD 13522-1, June 1993
- [JEF92] K. Jeffay, D. L. Stone, F. Donelson Smith, "Kernel support for live digital audio and video", Computer communications, Vol. 15, No. 6, July / August 1992
- [JEF94a] K. Jeffay, D.L. Stone, F.D. Smith, "Transport and display mechanisms for multimedia conferencing across packet-switched networks", Computer networks and ISDN systems, Vol. 26, 1994
- [JEF94b] K. Jeffay, "On latency management in Time-Shared Operating Systems", IEEE Newsletter on real time systems, 1994
- [KAA91] M.F. Kaashoek, A.S. Tanenbaum, "Group communication in the AMOEBA distributed system", Proceedings of the IEEE: 11th International Conference on Distributed Computing Systems, pages 222-230, May 1991
- [KAN94] Kang G. Shin, Parameswaran Ramanathan, "Real time computing : a new discipline of computer science engineering", Proceedings of the IEEE, Vol. 82, No. 1, January 1994
- [KAP92] S. M Kaplan, A. M. Caroll, "Supporting Collaborative Processes with Conversation Builder", Computer Communications, Vol. 15, No. 8, Octobre 1992
- [KAR90] A. Karmouch, L. Orozco-Barbosa, N. Georganas, M. Goldberg, "A multimedia communication system", IEEE journal on selected areas in communications, Vol. 8, No. 3, april 1990
- [KAR93] A. Karsenty, M. Baudoin-Lafon, "An algorithm for distributed groupware applications", Proceedings of the IEEE : 13th International conference on distributed computing systems, pages 195-202, May 1993
- [KAR94a] A. Karsenty, "Le collectifiel : de l'interaction homme-machine à la communication homme-machine-homme", Technique et science informatiques, Vol. 13, No. 1, pp 105-127, 1994
- [KAR94b] L. Karsenty, E. Drieux, "Vers un environnement de télé-tutorat : Analyse de la communication instructeur-élèves dans une salle de cours", Rapport Technique EURISCO N° T-94-011, 1994

- [KAT94] D.I. Katcher, K.A. Kettler, J.K. Strosnider, "Modeling DSP Operating Systems for Multimedia Applications", Proceedings of the Real Time Systems Symposium, San Juan, Puerto Rico, December 7-9, 1994
- [KEL94] R. K. Keller, X. Shen, G. v. Bochmann, "Macronet - A Simple yet Expressive and Flexible Formalism for Business Modelling", Workshop on Computer Supported Cooperative Work, Petri Nets and related formalisms, Saragoza, pages 51-55, June 1994
- [KLE96] S. Kleiman, D. Shah, B. Smaalders, "Programming with threads", 534 pages, Prentice Hall 1996
- [KNI90] M.J. Knister, A. Prakash, "DistEdit : a distributed toolkit for supporting multiple group editors", proceedings of the ACM SIGCHI & SIGOIS : 3rd conference on computer supported cooperative work, pages 343-355, October 1990
- [KRA91] H. Krasner, J. McInroy, D. B. Walz, "Groupware Research and Technology Issues with Application to Software Process Management", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 4, pp 704-712, July/August 1991
- [KRE89] T. Kreifelts, F. Victor, G. Woetzel, M. Weitass, "A Design Tool for Autonomous Group Agents", Studies in Computer Supported Cooperative Work: Theory, Practice and Design, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991
- [KUM91] A. Kumar, "Hierarchical Quorum Consensus: A New Algorithm for Managing Replicated Data", IEEE Transactions on Computers, Vol. 40, No. 9, pp 996-1004, September 1991
- [LEG91] D. Le Gall, "MPEG: a video compression standard for multimedia applications", Communications of the ACM, Vol. 34, No. 4, April 1991
- [LIA90] L. Liang, S. T. Chanson, G.W. Neufeld, "Process Groups and Group Communications: Classifications and Requirements", IEEE Computer, Vol. 23, No. 2, pp 56-66, February 1990
- [LIO91] M. Liou, "Overview of the px64 kbits/s video coding standard", Communications of the ACM, Vol. 34, No. 4, April 1991
- [LIT90] T.D.C. Little, A. Ghafoor, "Synchronization and storage models for multimedia objects", IEEE journal on selected areas in communications, Vol. 18, No. 3, april 1990
- [LIT93] T.D.C. Little, "A framework for synchronous delivery of time dependant multimedia data", multimedia systems, No. 1, 1993
- [MAR90] T. Maruichi, M. Ichikawa, M. Tokoro, "Modelling Autonomous Agents and their Groups", In Decentralized A.I., Eds. Y. Demazeau and J. P. Müller, North-Holland, 1990
- [MAR95] A.D. Marshall, "Developing Hypertext Courseware on the Worls Wide Web", Proceedings of ED'MEDIA 95, 1995
- [MAZ91] M. S. Mazer, "Reasoning About Knowledge to Understand Distributed AI Systems", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 6, pp 1333-1346, November/Déceber 1991
- [MCC94] S. McCanne, V. Jacobson, "VIC - Video Conderence", Unix manual pages, November 1994
- [MER74] P. Merlin, "A study of the recoverability of computer systems", thesis, Computer Science dept., University of California, Irvine, 1974
- [MEY88] B. Meyer, "Object-oriented software construction", Prentice Hall International, 1988

- [MIN93] S. Minör, B. Magnusson, "A model for Semi-(a)Synchronous Collaborative Editing", Proceedings of the Third European Conference on Computer Supported Cooperative Work, pages 219-231, Eds. G. de Michelis, C. Simone and K. Schmidt, Kluwer Academic Publishers, September 1993
- [MIT91] R. Mittman, "Dans cinq ans, toutes les applications seront des groupwares", Télécom Magazine, Vol. 9, pp 63-65, Octobre 1991
- [MUR93] J. Murray, "K12 Network : Global Education Through Telecommunications", Communications of the ACM, vol. 36, No. 8, august 1993
- [NAK95] K. Nakabayashi, Y. Koile, M. Maruyama, H. Toubai, S. Ishiuchi, Y. Fukuhara, "An Intelligent Tutoring System on World Wide Web : Towards an Integrated Learning Environment on a Distributed Hypermedia", Proceedings of ED'MEDIA 95
- [NEI92] M. L. Neilsen, M. Mizuno, M. Raynal, "A general Method to Define Quorums", Proceedings of the IEEE: 12th International Conference on Distributed Computing Systems, pages 657-664, June 1992
- [NGO91] L. H. Ngoh, "Multicast support for group communications", Computer Networks and ISDN systems, Vol. 22, No. 3, pp 165-178, October 1991
- [OHM92] T. Ohmori, K. Maeno, S. Sakata, H. Fukuora, K. Watabe, "Distributed Cooperative Control for Sharing Applications Based on Multiparty and Multimedia Desktop Conferencing System: MERMAID", Proceedings of the IEEE: 12th International Conference on Distributed Computing Systems, pages 538-546, Juin 1992
- [OIK94] S. Oikawa, H. Tokuda, "User-Level-Real-Time-Threads", IEEE Newsletter on Real Time Systems, 1994
- [OWE93] P. Owezarski, "Etude et Spécification de Systèmes de Communications de Groupes pour les Applications Coopératives", Rapport de stage de DEA Interface Homme-Système MultiModale (IHS2M), Université Paul Sabatier, Juin 1993
- [OWE95a] P. Owezarski, M. Diaz, P. Sénac, "Modélisation et implémentation de mécanismes de synchronisation multimédia dans une application de visioconférence", Actes du colloque francophone sur l'ingénierie des protocoles, (CFIP'95), pp 305-319, Editions Hermès, Rennes, France, 10-12 mai 1995
- [OWE95b] P. Owezarski, V. Baudin, M. Diaz, J.F. Schmidt, "Multimédia Teleteaching : Introduction of Synchronization and Cooperation Mechanisms in Distance Learning", Proceedings of the world conference on educational multimedia and hypermedia (ED'MEDIA 95), pp 517-522, Graz, Austria, June 17-21, 1995
- [OWE95c] P. Owezarski, M. Diaz, "Gestion de la synchronisation multimédia et de la qualité de service dans une application de visioconférence : mécanismes système et transport", Rapport LAAS No. 95460, 62 pages, Novembre 1995
- [OWE96a] P. Owezarski, T. Villemur, M. Diaz, "Conception d'un système de visioconférence coopératif", Actes des 8ème rencontres francophones sur le parallélisme (RenPar'8), pages 25-28, Bordeaux, France, 20-24 Mai 1996
- [OWE96b] P. Owezarski, T. Villemur, M. Diaz, "Conception et implémentation d'un système de visioconférence coopératif à N intervenants", Actes des journées de recherche sur le contrôle réparti dans les applications coopératives (CRAC'96), pages 87-92, Paris, France, 30-31 Mai 1996

- [OWE96c] P. Owezarski, M. Diaz, "Models for enforcing multimedia synchronization 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
- [OWE96d] P. Owezarski, T. Villemur, M. Diaz, "Conception et implémentation d'un système de visioconférence coopératif", Rapport LAAS N° 96401, Août 1996
- [OWE96e] P. Owezarski, M. Diaz, "Models for enforcing multimedia synchronization and QoS in visioconference applications", LAAS report No 96287, 16 pages, July 1996
- [OWE96f] P. Owezarski, V. Baudin, M. Diaz, "Conception et réalisation d'un environnement de téléformation synchrone", Rapport LAAS No 96285, 27 pages, Juillet 1996
- [OWE96g] P. Owezarski, V. Baudin, M. Diaz, "Conception et développement d'un système synchrone de téléformation professionnelle", Actes des séminaires hypermédia, éducation et formation, Paris, France, 22 mars 1996
- [PAQ95] G. Paquette, C. Ricciardi-Rigault, J. Bourdeau, C. Paquin, S. Liégeois "Modeling a Virtual Campus Environment for Interactive Distance Learning", Proceedings of ED'MEDIA 95, 1995
- [PAR94] Parallax Graphics Inc., "Software Application Guide : Video & Multimedia Applications", 1994
- [PER96] M.J. Pérez-Luque, T.D.C. Little, "A temporal reference framework for multimedia synchronization", IEEE journal on selected areas in communications, Vol. 14, No. 1, January 1996
- [POS80] J. Postel, "User Datagram Protocol (UDP)", Request For Comments No. 768, August 1980
- [POS82] J. Postel, "Simple Mail Transfert Protocol (SMTP)", Request For Comments No. 821, August 1982
- [POW91] M.L. Powell, S.R. Kleiman, S. Barton, D. Shah, D. Stein, M. Weeks, "SunOS Multi-thread Architecture", USENIX, Dallas, TX, Winter'91
- [POZ94] S. Pozzi, "An Integrated Environment for Distance Education Supporting Multiple Interaction Styles", Proceedings of ED'MEDIA 94, 1994
- [POZ95] S. Pozzi, "Supporting Synchronous and Asynchronous Tele-Tutoring Activities by Using T.E.M.P.O.", Proceedings of ED'MEDIA 95, 1995
- [PRI89] W. Prinz, P. Pennelli, "Relevance of the X.500 Directory to CSCW Applications", Studies in Computer Supported Cooperative Work: Theory, Practice and Design, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991
- [REZ96] J.F. de Rezende, S. Fdida, A. Mauthe, D. Hutchinson, "Scalable and reliable multimedia multipeer communication in the M-connection service", IEEE journal on selected areas in communications, Vol. 14, No. 1, January 1996
- [ROB94] J. A. Robinson, "Communications Services Architecture for CSCW", Computer Communications, Vol. 17, No. 5, pp 339-347, Mai 1994
- [ROD92] T. Rodden, G. S. Blair, "Distributed Systems Support for Computer Supported Cooperative Work", Computer Communications, Vol. 15, No. 8, pp 527-538, October 1992

- [ROT95] K. Rothermel, T. Helbig, "An Adaptive Stream Synchronization Protocol", proceedings of the Network and Operating System Support for Digital Audio and Video conference (NOSSDAV'95), 1995
- [RUS94] L. F. Rust Da Costa Carmo, "Méthodes de synchronisation dans les systèmes répartis multimédia: une approche intégrant relations de causalité et contraintes temporelles", Thèse de doctorat de l'université Paul Sabatier (Toulouse III), Novembre 1994
- [SAR85] S. Sarin, I. Greif, "Computer-based real-time conferencing systems", IEEE computer, pages 33-45, october 1985
- [SCH93] A. Schiper, A. Sandoz, "Uniform Reliable Multicast in a Virtually Synchronous Environment", Proceedings of the IEEE: 13th International Conference on Distributed Computing Systems, pages 561-568, May 1993
- [SCH96a] J. Schnepf, J.A. Konstan, D. Hung-Chang Du, "Doing FLIPS : Flexible Interactive Presentation Synchronization", IEEE journal on selected areas in communications, Vol. 14, No. 1, January 1996
- [SCH96b] Schlzrinne et al., "RTP: A Transport Protocol for Real-Time Applications", Request For Comments N° 1889, January 1996
- [SEN94] P. Sénac, M. Diaz, P. De Saqui-Sannes, "Toward a formal specification of multimedia synchronization", Annals of telecommunications, 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, January 1996
- [SMI81] R. G. Smith, R. Davis, "Frameworks for Cooperation in Distributed Problem Solving", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 11, No. 1, pp 61-70, January 1981
- [SMI89] H. Smith, P. Hennessy, G. Lunt, "An Object-Oriented Framework for Modelling Organisational Communication", Studies in Computer Supported Cooperative Work: Theory, Practice and Design, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991
- [STE90] R. Steinmetz, "Synchronization properties in multimedia systems", IEEE journal on selected areas in communications, Vol. 8, pp 401-412, April 1990
- [STE92] D. Stein, D. Shah, "Implementing Lightweight Threads", Summer'92 USENIX, San Antonio, TX, June 8-12, 1992
- [SUN94a] SunSoft, Solaris 2.4 Answerbook, "Multithreaded Programming Guide", 1994
- [SUN94b] SunSoft, Solaris 2.4 Answerbook, "System Services Guide : Chapter 4 : Process Scheduler", 1994
- [SUN94c] SunSoft, Solaris 2.4 Answerbook, "System Services Guide : Chapter 6 : Realtime Programming and Administration", 1994
- [TOU94] P. Touzeau, "Formalism for Documents Circulation Application", Workshop on Computer Supported Cooperative Work, Petri Nets and related formalisms, Saragoza, pages 1-9, June 1994
- [TUR93] T. Turletti, "H.261 software codec for videoconferencing over the internet", Rapport de recherche INRIA No. 1834, Janvier 1993
- [TUR95] T. Turletti, "Contrôle de transmission pour logiciel de vidéoconférence", Thèse de doctorat, 1995
- [VER95] F. Vernadat, T. Villemur, P. Gradit, M. Diaz, "Distributed management of dynamic membership of agents in cooperative groups", 15th international conference on application and theory of Petri nets, European Research Seminar on Advances in Distributed Systems editor, Pages 236-241, L'Alpe d'Huez, 1995

- [VIC89] F. Victor, E. Sommer, "Supporting the design of Office Procedures in the DOMINO System", *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991
- [VIL95] T. Villemur, "Conception de services et de protocoles pour la gestion de groupes coopératifs", Thèse de doctorat de l'université Toulouse III, Janvier 1995
- [VOG95] A. Vogel, B. Kerhervé, G. Von Bochmann, J. Gecsei, "Distributed Multimedia and QoS : A Survey", *IEEE Multimedia*, Summer 1995
- [WAL83] B. Walter, "Timed Petri-nets for modeling and analysing protocols with time", *proceedings of PSTV, III, IFIP*, 1983
- [WAL91] G.K. Wallace, "The JPEG picture compression standard", *Communications of the ACM*, Vol. 34, No. 4, April 1991
- [WIL91] P. Wilson, "Computer Supported Cooperative Work (CSCW): origins, concepts and research initiatives", *Computer Networks and ISDN Systems*, Vol. 23, No. 1-3, pp 91-95, November 1991
- [WIL94] J.M. Wilson, D.N. Mosher, "Interactive MultiMedia Distance Learning (IMDL) : The prototype of the Virtual Classroom", *Proceedings of ED'MEDIA 94*, 1994
- [WOL95] K.D. Wolf, "The implementation of an open learning environment under World Wide Web", *Proceedings of ED'MEDIA 95*, 1995
- [ZLO91] G. Zlotkin, J. S. Rosenschein, "Cooperation and Conflict Resolution via Negotiation Among Autonomous Agents in Noncooperative Domains", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 6, pp 1317-1324, November/Décember 1991

Thèse de Philippe Owezarski

**Conception et formalisation d'une application de visioconférence coopérative.
Application et extension pour la téléformation**

Résumé :

Les progrès récents dans le domaine de l'informatique et des réseaux de communications ont ouvert la voie aux applications distribuées multimédias coopératives. La problématique associée à la conception de telles applications comporte plusieurs points. Tout d'abord, les données multimédias se caractérisent par leur qualité de service en termes de fiabilité, de débit engendré, de synchronisation temporelle... Les applications multimédias doivent donc garantir le respect de la qualité de service de chacun des médias, le point essentiel consistant à assurer le respect des contraintes de synchronisation intra et inter-flux. De même, le support de communication doit pouvoir s'adapter aux contraintes des médias transportés et fournir un service adéquat en terme de débit, fiabilité, délai de bout en bout... Enfin, les utilisateurs ont de plus en plus besoin de travailler en groupe pour leurs applications informatiques, et des mécanismes de coopération doivent être introduits.

Dans cette thèse, des mécanismes ont été proposés pour répondre à cette problématique et ont été mis en oeuvre dans le cadre d'une application de visioconférence. Ainsi, la synchronisation multimédia est réalisée par un moteur utilisant des mécanismes avancés des systèmes opératoires et respectant un scénario modélisé par un réseau de Petri à flux temporels. La communication utilise un transport à ordre partiel qui s'adapte bien aux contraintes de cette application autant en terme de débit que de fiabilité, et permet d'en augmenter les performances. Enfin, cette visioconférence a été étendue pour prendre en compte des notions de travail de groupe et offre, en particulier, un contrôle des entrées/sorties des participants et des interactions au sein du groupe. Une architecture générale garantissant ces contraintes temporelles et de coopération a ainsi été proposée et réalisée.

Enfin, ces techniques ont été appliquées à une application de téléformation professionnelle dans le domaine de l'aéronautique.

Mots-clés : multimédia, synchronisation, système opératoire, transport à ordre partiel, coopération, communications de groupe, visioconférence, téléformation

**Formal design of a cooperative videoconference application.
Application and extension for tele-training**

Abstract:

Recent progress in computer science and networking are able to support cooperative distributed multimedia applications. The set of problems related to the design of such applications consists of several points. First, multimedia data is characterized by its quality of service in terms of reliability, throughput, temporal synchronization,... Multimedia applications have to ensure the quality of service of each of the medias, the key point being to ensure intra and inter-streams synchronization constraints. Also, the communication system has to adapt itself to the constraints of the transported medias and to provide a suitable service in terms of throughput, reliability, end to end delay... Finally, users more and more require to jointly share applications, and cooperation mechanisms have to be introduced.

In this thesis, mechanisms have been proposed to provide solutions for these problems and have been used in a videoconference application. Then, multimedia synchronization is performed by a synchronization software that uses advanced operating systems mechanisms and conforms to a scenario modeled by a time stream Petri net. Communication uses a partial order transport that adapt itself to the application constraints in terms of throughput, reliability and allows the application performances to be improved. Finally, this videoconference application has been extended to take into account the workgroups and provides, in particular, a control of the joining/leaving participants of the group. A general architecture ensuring these temporal and cooperative constraints has been proposed and realized. Finally, these mechanisms are shown to be applicable to a professional tele-training application in aeronautics.

Keywords : multimedia, synchronization, operating system, partial order transport, cooperation, group communications, videoconference, tele-training