

Chapitre \$

PROTOCOLES DE TRANSPORT MULTIMEDIAS

MARC BOYER, CHRISTOPHE CHASSOT, LAURENT DAIRAINÉ, MICHEL DIAZ, ANDRÉ
LOZES, PHILIPPE OWEZARSKI, LUIS ROJAS-CARDENAS

\$.1. Introduction

Ces dernières années ont vu une avancée fulgurante dans l'informatique multimédia et les transmissions à haute vitesse, ce qui *constitue le fait marquant de la fin de ces années 90.*

De tels sauts technologiques sous-tendent de nouveaux problèmes et de fortes innovations, en particulier au niveau des architectures et des applications. Ainsi, de nombreux travaux fondamentaux sont encore indispensables pour :

- le développement des réseaux hauts débits,
- la création de nouvelles applications,

2 Titre de l'ouvrage

- l'étude de nouveaux outils et méthodes de support,
- l'expérimentation sur des plates-formes avancées.

Les débits proposés aux utilisateurs passent de l'ordre des *Kilobit/s* à celui des *Megabit/s* et bientôt des centaines de *Megabit/s*. De plus, de nouveaux réseaux alternatifs à hauts débits apparaissent, à la fois dans le monde des liaisons sans fil et des liaisons satellitaires pendant que les *informations multimédias* se développent. La complexité résultante peut facilement être appréhendée si l'on note qu'une vidéo T.V. - Haute Définition nécessite des débits isochrones de l'ordre de *1 Gigabit/s* en mode non compressé et de *10 Megabit/s* en mode compressé.

Ces nouvelles possibilités bousculeront dans les prochaines années les réseaux, ce qui explique l'importance de l'initiative américaine concernant *l'Internet nouvelle génération et les réactions européennes*. Il faut *remplacer les logiciels communicants existants par ceux de la nouvelle génération*. Sophistiqués, ils devront supporter à un niveau mondialement distribué, sur tous les supports physiques, les applications du futur.

Les études à développer devant répondre à un certain nombre de critères, nous allons maintenant présenter rapidement les principaux problèmes qui se posent lors de la conception de tels systèmes, par leurs aspects réseaux et architectures.

§.1.1. Les informations multimédias

Une communication multimédia interconnecte des participants échangeant des données diverses (voix, données, images vidéo) comportant des contraintes temporelles. L'intégration dans les réseaux de ces différents médias pose des problèmes nouveaux, car : la nature même des informations communiquées, telle que la vidéo, nécessite par utilisateur des débits importants non encore disponibles. De plus, plus que les débits, c'est l'hétérogénéité des caractéristiques temporelles de transmission de ces informations multimédias qui introduit des contraintes nouvelles en terme de qualité de service et de synchronisation.

Ainsi, après la première génération actuelle, il faut maintenant concevoir les nouveaux mécanismes, protocoles, services et logiciels haute vitesse multimédias, permettant d'atteindre, en préservant toute la cohérence temporelle des flux, des débits de l'ordre de la centaine de *Megabit/s* pour des ensembles d'utilisateurs coopérants.

§.1.2. L'architecture

Actuellement l'Internet ne fournit aucune garantie de qualité de service, c'est-à-dire que les vitesses et les performances obtenues par les utilisateurs dépendent totalement de la charge en communication du réseau, ceci au moment où la communication est effective. La solution réseau future, celle qui est en cours d'étude, doit ainsi en premier lieu posséder à la fois les intérêts informatiques de l'Internet et les intérêts de qualité de service des réseaux téléphoniques : il faut assurer à tous les utilisateurs l'augmentation simultanée de leurs débits, de leur nombre de correspondants, et des fonctionnalités qui leur sont offertes car les nouveaux réseaux pourront comporter des dizaines de milliers d'équipements, du simple téléphone classique ou de l'équipement domotique de base, à l'ordinateur parallèle ou au serveur le plus sophistiqué.

Cette croissance implique qu'il devient indispensable, afin d'assurer la qualité de la coopération globale, de fournir une qualité de communication définie et certifiée.

La conception et le développement de ces nouveaux services et protocoles actuels peuvent être classés selon deux approches :

- l'approche "Applications orientées Réseaux" considère que les applications doivent s'adapter au mieux et si possible parfaitement aux problèmes et aux fluctuations du réseau [CLARK 90] [DIOT 95b] ;
- la deuxième, "Réseaux orientés Applications" reconçoit les réseaux en y incluant toutes les caractéristiques nécessaires aux nouvelles applications [AMER 94] [DIAZ 95].

Ces deux approches, qui n'ont pas encore été complètement évaluées, sont à la base de la conception de la prochaine génération des protocoles et des applications distribuées.

La deuxième approche, plus générale et générique, sera celle développée ici. On montrera comment, à partir d'une explicitation des nouveaux besoins, et plus précisément d'une description explicite précise des objets multimédias les plus généraux, elle conduira à la conception d'une nouvelle architecture et de nouveaux protocoles particulièrement adaptés aux systèmes multimédias distribués.

§.1.3. La Qualité de service (QoS), la deuxième génération de l'Internet

Il faut souligner que, dans le contexte le plus général, garantir une QoS donnée s'avère un problème très difficile dans une grande architecture distribuée. En effet, de nouveaux mécanismes et de nouvelles solutions sont nécessaires car :

- les flux transportés deviennent moins sensibles aux erreurs logiques,
- les besoins évoluent vers les applications à temps contraint,

4 Titre de l'ouvrage

- le contrôle d'erreur devient un paramètre variable de la conception,
- les protocoles doivent être allégés sans perte de fonctionnalité,
- des mécanismes de contrôle d'accès, de flux, de débit et de congestion doivent permettre de satisfaire les contraintes temporelles.

Les protocoles actuels de type TCP devenant mal adaptés à ces nouveaux besoins et ceux de type UDP n'étant pas suffisants, les solutions et réalisations devront traiter simultanément tous ces nouveaux problèmes. Pour nous, les logiciels devront manipuler et assurer le transport des objets multimédias les plus généraux, en isolant leurs composantes et en assurant leur synchronisation (donc en les décomposant, les transférant et les recomposant, car ces objets seront distordus lors du transfert). Il faut assurer le maintien de l'isosynchronisme des flux à travers les réseaux, la maîtrise des délais, des gigue et des dérives intra-flux et inter-flux, des protections et des priorités.

Dans notre approche, la notion de garantie est fondamentale, quelles que soient les contraintes liées à celle-ci. Plus ou moins fortes, les contraintes de QoS devront être précisées, et en tout cas une ou des bornes minimales ou maximales seront précisées, et la conception devra les satisfaire.

Afin de garantir cette QoS, notre conception, et c'est une originalité, repose sur une base formelle, qui devient fondamentale. Nous partirons d'un modèle des objets multimédias, et à partir de ce modèle, nous en déduirons une architecture originale parfaitement adaptée à ces applications.

En conséquence, ce chapitre présentera d'une part l'approche suivie, le modèle multimédia utilisé et d'autre part la nouvelle architecture et les nouveaux protocoles obtenus par cette démarche originale. Enfin, des exemples montreront l'intérêt à la fois des applications réalisées et des fonctionnalités obtenues.

§.2. Les protocoles de Transport actuels

Le rôle de la couche Transport de toute architecture de réseau commuté (ou d'interconnexion de réseaux tel que l'Internet) est « d'émuler », pour toute instance d'application distribuée, la disponibilité d'un canal de communication de bout en bout :

- dédié aux échanges de données entre les programmes applicatifs considérés ;
- bidirectionnel, c'est à dire permettant à ses utilisateurs d'échanger des données dans les deux sens ;
- offrant comme QoS : soit ordre total et fiabilité totale (ni erreur, ni perte, ni duplicata), soit ni ordre, ni fiabilité .

Pour cela, la couche Transport dispose du (des) service(s) offert(s) par le réseau sous-jacent, le plus généralement un réseau commuté (voir le chapitre ? pour une présentation plus détaillée des fonctions et mécanisme de la couche transport).

En plus des problèmes de panne (rupture d'un lien ou défaillance d'un nœud intermédiaire par exemple), on distingue deux principaux types d'imperfections potentielles dans un réseau commuté :

- le « déséquencement » des paquets au travers du réseau, conséquence possible d'une commutation de ces paquets « en mode datagramme » ;
- la « pertes de paquet(s) », conséquence possible de deux sortes d'événements : le rejet d'une trame sur détection d'erreur(s) bit, ou le rejet d'un paquet en raison de la congestion d'un commutateur intermédiaire.

Notons qu'indépendamment du réseau sous-jacent, des pertes peuvent également intervenir par congestion du hôte destinataire, lorsque (par exemple), le programme applicatif récepteur lit « moins vite » les données que ne les envoie le programme émetteur.

Enfin, deux contraintes supplémentaires sont éventuellement imposées par le réseau :

- une limite sur la taille des données soumises par la couche Transport, fonction de la MTU (*Maximum Transmission Unit*) du réseau ;
- un nombre limité de points d'accès au service de niveau Réseau.

§.2.1. UDP et TCP

On distingue deux protocoles de Transport majeurs dans l'Internet : TCP [RFC 793] et UDP [RFC 1323], implantés dans toute machine hôte d'un réseau connecté à l'Internet. Avant d'en étudier les mécanismes principaux, résumons les caractéristiques générales de l'Internet et plus précisément celles du protocole d'interconnexion des différents réseaux de l'Internet : le protocole IP (*Internet Protocol*).

Le rôle principal du protocole IP est d'assurer le routage de « paquets IP » d'un hôte émetteur vers un ou plusieurs hôtes récepteurs, sans aucune garantie d'ordre ni de fiabilité. Pour cela, IP met en œuvre une technique de commutation de paquets en mode datagramme, chaque paquet IP comportant « l'adresse IP » des machines source et destinataire, et étant géré de façon indépendante des autres paquets. En outre, IP assure la segmentation et le réassemblage des données trop grandes au regard de la MTU des réseaux traversés.

§.2.1.1. TCP

TCP (*Transmission Control Protocol*) est un protocole avec connexion conçu pour fournir un service de transfert de données de bout en bout, en mode flux d'octets, fiable et ordonné, et ce dans un environnement réseau de type Internet.

En conséquence, les mécanismes principaux de TCP sont les suivants : établissement et libération des connexions, multiplexage et démultiplexage, reprise des pertes, contrôle de flux. Notons que dans l'Internet, les adresses de niveau Transport (c'est à dire identifiant chacun des processus applicatif communicant) sont composées de l'adresse IP de la machine sur laquelle s'exécute le processus, et d'un « numéro de port » identifiant le point d'accès au service créé et utilisé par le processus (« socket ») parmi tous ceux créés localement. En conséquence, la conversion d'une adresse Transport en adresse Internet est un mécanisme trivial. Notons également que TCP autorise le transfert de données urgentes, c'est à dire non soumises au contrôle de flux.

Outre ces mécanismes, TCP met en œuvre un mécanisme fondamental pour le bon fonctionnement de l'Internet : le contrôle de congestion (algorithme du « Slow Start »). Ce mécanisme a pour objectif de diminuer le volume des émissions d'un hôte lorsque intervient la congestion d'un routeur IP intermédiaire, cette congestion étant détectée grâce à l'expiration du « timer » associé à un PDU TCP émis et non encore acquitté, ou par la réception de 3 ACK dupliqués.

§.2.1.2. UDP

UDP (*User Datagram Protocol*) est un protocole sans connexion conçu pour fournir un service de transfert de données de bout en bout, en mode message, sans garantie d'ordre ni de fiabilité, dans un environnement réseau de type Internet. En conséquence, le seul véritable mécanisme de UDP est d'assurer le multiplexage et le démultiplexage des canaux UDP sur l'unique canal IP.

§.2.2. Nouveaux besoins

Initialement conçus pour assurer le transfert de données textuelles sur des réseaux peu fiables ou à bas débit, les protocoles de Transport actuels s'avèrent inadaptés aux nouveaux contextes applicatif et réseau. En effet, leurs fonctionnalités ne répondent ni aux besoins des applications multimédias (hauts débits, délais de transit bornés et synchronisation notamment), ni à la volonté d'optimiser les ressources de communication dans les réseaux de distribution.

Ces dernières années, de nombreux travaux de recherche ont porté sur la conception de nouveaux protocoles de Transport « haute vitesse », destinés d'une part à être mis en œuvre dans des environnements réseaux à hauts débits, et d'autre part à supporter le transfert de gros volumes de données. Ces évolutions ont en

particulier conduit à définir des protocoles moins complexes offrant une fiabilité paramétrable par leurs utilisateurs, ainsi qu'un mécanisme de contrôle de débit [CLARK 87], [CHERINTON 89], [PEI 92].

Cependant, aucune de ces évolutions ne permet de prendre en compte les exigences spécifiques des applications multimédias au niveau de la couche Transport, en particulier la synchronisation (spatiale ou temporelle) des flux multimédias tels que l'audio ou la vidéo. En cela, elles n'abordent la problématique du multimédia distribué qu'en terme de hauts débits, laissant aux utilisateurs du service le soin de gérer l'ensemble des contraintes de synchronisation.

Dans ce contexte, plusieurs propositions d'architecture de communication « orientée multimédia » ont été formulées, parmi lesquelles celle constituant l'objet de ce chapitre. Toutes visent à prendre en compte et à satisfaire les caractéristiques et les contraintes des applications multimédias, mais diffèrent (en particulier) pour ce qui concerne les fonctionnalités de la couche Transport.

Deux approches majeures se sont opposées ces dernières années dans la conception de ces architectures.

La première approche préconise (ALF) une gestion des contraintes applicatives par l'application elle-même. En cela, elle requiert de la part du service de Transport un minimum de QoS (UDP convient). Cette approche repose sur l'hypothèse que l'application est apte à s'adapter aux fluctuations du réseau [DIOT 95a], [BOLOT 96], [BOLOT 98]. Clairement, son coût réside dans l'accroissement de la complexité des logiciels applicatifs qu'elle engendre. Notons cependant que des investigations visant à diminuer cette complexité ont été menées dans le cadre du projet Esprit HIPPARCH, et auxquelles fait référence [DIOT 95b].

A l'opposé, la deuxième approche propose d'accroître la complexité des services et protocoles de Transport de façon à prendre en compte les besoins applicatifs. L'application se voit ainsi offrir un service de Transport lui permettant, via une API étendue, de spécifier ses besoins au moyen de paramètres de QoS significatifs de son point de vue. L'objectif de cette approche est de faciliter le développement des logiciels applicatifs, au prix d'une sophistication de la couche Transport. Cette approche a été suivie au travers de plusieurs projets : RACE CIO [HENCKEL 93], BERKOM [DELGROSSI 93], OSI95 [DANTHINE 94], TENET [TENET 94], QoS-A [CAMPBELL 94] ou encore CESAME en France [DIAZ 94a].

Les contributions présentées en section §.4 et §.5 de ce chapitre se situent dans le cadre de cette deuxième approche. Plus spécifiquement, elles consistent en la proposition d'une nouvelle architecture de Transport utilisant une extension de la

notion classique de connexion : la « connexion d'ordre partiel et de fiabilité partielle. »

Avant d'étudier plus en détail cette architecture, nous introduisons dans la section suivante les différents modèles de représentation de la représentation de la synchronisation dans les applications multimédias, et en particulier le modèle TSPN, que nous réutiliserons ultérieurement.

§.3. Les modèles des informations multimédias

Avant de décrire la notion de connexion à ordre partiel, il est important de bien connaître l'ensemble des paramètres et attributs – que l'on regroupe sous le terme QdS – que doit respecter une connexion. La notion de QdS reste très abstraite et d'ailleurs chaque couche de la pile de communication en possède sa propre définition. Aussi, il est indispensable de parfaitement exprimer tous les besoins et toutes les contraintes à chaque niveau. Pour cela, la solution consiste à utiliser un formalisme adéquat, suffisamment puissant en terme de modélisation et de pouvoir d'expression¹, de façon à exprimer de façon précise et complète les besoins en QdS de chaque niveau.

§.3.1. Importance de la modélisation des contraintes de synchronisation et de QdS

Du point de vue des utilisateurs, la complexité des scénarios multimédias réalistes (présentation, cours, etc.) est devenue telle que leur conception doit être assistée par un outil formel. Cette approche formelle est particulièrement intéressante car elle permet d'une part de spécifier sans ambiguïté des scénarios de présentations multimédias, et d'autre part de les simuler et de les valider. De nombreuses études ont déjà été réalisées dans ce domaine et des modèles ont été proposés [DIAZ 93a]. 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 œuvre des paradigmes multimédias tels que le paradigme de régie numérique [SENAC 94]. La section 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 modélisation

¹Le 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.

§.3.2. *Les modèles existants*

Nous ne présentons dans cette partie que les principaux modèles basés sur le formalisme des réseaux de Petri. D'autres approches existent, basée sur les notions d'intervalle [ALLEN 83], d'axe temporel [HOD89], de point de référence [STE90], ou fondées sur des solutions mixtes. Pour une présentation plus complète des différents modèles de modélisation multimédia, on pourra consulter [DIAZ 93b].

Nous mentionnerons tout particulièrement le modèle OCPN (Object Composition Petri Net) [LITTLE 90] qui utilise des réseaux de Petri en représentant chaque donnée (image, son, texte) par une place à laquelle on associe une durée. Sept schémas de synchronisation permettent d'exprimer les positions temporelles relatives des différentes données. Ce modèle cependant ne permet pas d'exprimer la gigue admissible intrinsèque à chaque objet.

Le modèle Arc Time Petri Net (ATPN) [WALTER 83], inspiré du modèle Time Petri Net (TPN) de Merlin [BERTHOMIEU 91], permet lui d'exprimer cette gigue temporelle. En effet, il place des 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 $\tau_{ij} + \beta_{ij}$

Ce modèle cependant est assez pauvre en ce qui concerne l'expression de synchronisation entre médias : sa seule règle consiste à fusionner des transitions. Une transition ne peut donc tirer que si toutes les contraintes de ces places incidentes sont satisfaites. Cette solution simple n'est toutefois pas adaptée aux objets multimédias, dans lesquels la gestion des décalages inter-flux est cruciale [DIAZ 93a].

§.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 Petri), appelé TSPN (Time Streams Petri Net), offrant à la fois le pouvoir d'expression du modèle TPN et le pouvoir de

modélisation du modèle ATPN. Le modèle TSPN [DIAZ 93a][DIAZ 93b] [SENAC 94] s'inspire du modèle ATPN auquel il ajoute le typage des transitions par des règles de tir de transitions inter-flux.

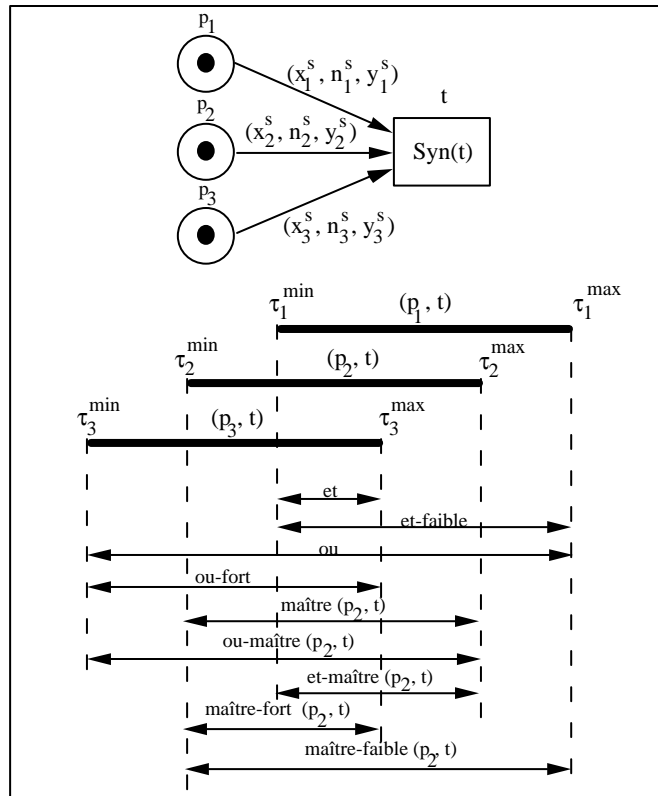


Figure \$1. Règles de tir des transitions inter-flux d'un TSPN

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).

Les dérives temporelles entre flux multimédias peuvent être contrôlées de façon très précise grâce à neuf 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 [DIAZ 93a] [SENAC 94]. 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 §.1, 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 respectivement $\tau_i + x_i^S$ et $\tau_i + y_i^S$.

§.3.4. Utilisations du modèle TSPN

Le modèle TSPN, tel qu'il est défini, est un modèle pour la conception de scénarios multimédias [WILLRICH 96], et il est utilisé comme tel pour la conception d'applications multimédias distribuées (cf. section §.6). Toutefois, dans le cadre de cet ouvrage, ce modèle est particulièrement intéressant car il ne se cantonne pas à la description des scénarios multimédias, mais il est également capable, par un modèle dérivé, de configurer le système de communications et sa couche Transport, pour que le service soit le plus adapté aux besoins de cette application (cf. sections §4 et §.5). Notamment, ce modèle met en évidence les relations de séquence et de parallélisme entre les objets composants le document multimédia et les contraintes de synchronisation entre les flux, ce qui, associé à la notion de fiabilité sur ces flux (paramètre de QoS variable suivant les médias et les besoins des utilisateurs) amène à la conception de nouveaux protocoles de communication, adaptés au multimédia, reposant sur des notions d'ordre et de fiabilité partiels. Notons que ces notions essentielles sont absentes des protocoles actuels de l'Internet TCP et UDP, les rendant du même coup mal adaptés aux nouveaux besoins de communication.

§.4. Les concepts d'ordre et de fiabilité (OF) multimédias

§.4.1. Les liens entre UDP et TCP

Tel que nous l'avons rappelé en section (§.2), les protocoles de Transport les plus utilisés à l'heure actuelle sont fondés sur l'alternative conceptuelle suivante :

- soit ils sont avec connexion (tels que TCP) et assurent à leurs utilisateurs un transfert de données sans déséquence, exempt d'erreur, de perte et de duplicata ; en cela, ces protocoles peuvent être qualifiés de « fiables et ordonnés »;
- soit ils sont sans connexion (tels que UDP) et ne garantissent ni le séquence, ni la fiabilité du transfert des données, celles-ci étant traitées en tant que datagrammes indépendants les uns des autres ; en cela, ces protocoles peuvent être qualifiés de « non fiables et sans ordre ».

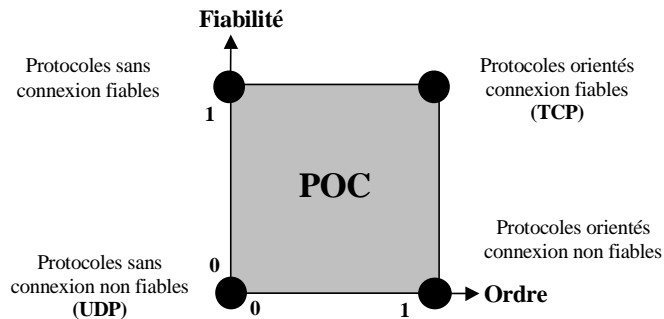


Figure §.2. Espace des protocoles d'ordre partiel et de fiabilité partielle

Cette classification souligne un vide conceptuel entre les deux familles de protocoles, suggérant une extension du concept de connexion en terme d'ordre et de fiabilité (figure §.2). Cette extension, la « connexion d'ordre partiel et de fiabilité partielle » (*POC: Partial Order Connection*), a été introduite et développée dans [AMER 94] et [DIAZ 94b]. Nous la présentons maintenant avant d'en montrer l'intérêt pour le transport des flux multimédias.

§.4.2. POC : une famille de nouveaux protocoles

§.4.2.1. Le concept de connexion d'ordre partiel et de fiabilité partielle

Une POC est une connexion de Transport permettant de définir et de mettre en œuvre tous les services d'ordre partiel et de fiabilité partielle entre deux entités communicantes. Concrètement :

- les données véhiculées au travers d'une POC peuvent être remises à l'utilisateur récepteur selon une séquence différente de celle de soumission par l'utilisateur

émetteur, à condition qu'elle respecte un ordre partiel particulier, échangé en tant que paramètre de QoS lors de l'établissement de la connexion ;

- de façon analogue, ces mêmes données peuvent ne pas être remises à l'utilisateur récepteur si les pertes correspondantes sont acceptables pour celui-ci ; au même titre que l'ordre partiel, la fiabilité partielle représente un paramètre de QoS, échangé lors de l'établissement de la connexion.

Notons que ce nouveau concept étend et unifie les deux approches traditionnelles de la notion de connexion représentées par UDP et TCP, qui apparaissent alors comme deux cas extrêmes de cette proposition.

§.4.2.2. POC : un support adapté au transfert des flux multimédias

L'intérêt d'un service de Transport garantissant une fiabilité partielle est facile à comprendre. Tout d'abord, le paramètre « fiabilité » est relativement simple à exprimer pour l'utilisateur du service : il peut par exemple s'exprimer en terme de pourcentage maximum de pertes acceptables, ou en nombre maximum de pertes consécutives acceptables. Par ailleurs, il permet aux utilisateurs de spécifier et d'obtenir toutes les garanties possibles en termes de fiabilité, depuis l'absence de garantie jusqu'à une garantie totale : en cela, un tel service permet de prendre en compte, au niveau Transport, les différences de QoS requise en terme de fiabilité pour le transfert du texte, du son ou de l'image (fixe ou animée) par exemple.

En terme de protocole, la mise en œuvre d'une fiabilité non plus totale mais partielle permet naturellement de dégager des bénéfices en terme de mémoire, de délai de transit, de mémoire et/ou de bande passante, par le fait qu'elle diminue (par exemple) le nombre de retransmissions de PDU, perdus au travers du réseau.

L'intérêt d'une garantie d'ordre partiel est plus subtil à concevoir. Nous l'illustrons au travers de l'exemples suivant.

EXEMPLE : cours d'anatomie d'une station serveur vers une station cliente.

Extrait de [LITTLE 90], l'exemple considéré consiste en la distribution d'un cours d'anatomie d'une station serveur vers une station cliente (figure §.3).

Sept fenêtres figurent à l'écran de la station cliente.

- Deux icônes représentant respectivement un organe (par exemple le cœur) et la localisation de cet organe dans le corps humain figurent sur deux fenêtres ; ils sont remis à jour à chaque visualisation d'un nouvel organe.
- Des commentaires textuels sur l'organe sont affichés sur deux autres fenêtres.
- Sur deux autres fenêtres apparaissent des séquences d'images fixes présentant l'organe étudié en rotation autour d'un axe suivant deux angles différents.

- Une séquence vidéo présentant différentes courbes (telles que les variations du nombre de pulsations cardiaques au cours du temps, si l'organe visualisé est le cœur) apparaît sur une dernière fenêtre.
- Enfin, un commentaire audio des courbes précédentes est diffusé au fur et à mesure de leur évolution dans le temps.

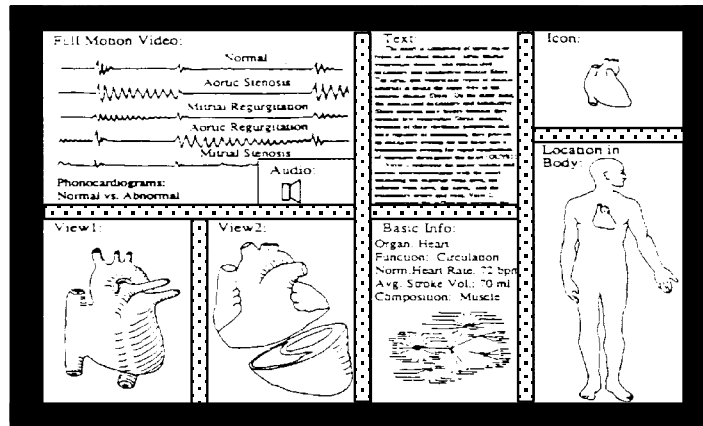


Figure 3.3. Distribution d'un cours d'anatomie[LITTLE 90]

Les fenêtres ne se recouvrant pas, l'application présente un parallélisme de présentation analogue à celui du cas (d) de l'exemple précédent. Cependant, il apparaît ici un nouveau type de contraintes lié à la chronologie de présentation des différents médias.

Ces contraintes expriment par exemple que :

- les images des deux séquences d'images fixes doivent être restituées dans leur ordre d'acquisition ;
- ces mêmes images doivent être présentées de façon à correspondre au même cliché instantané (sous deux angles différents) de l'organe en rotation ;
- les icônes et les commentaires textuels doivent être à tout moment relatifs aux séquences d'images fixes qui s'affichent à l'écran.

Considérons alors le réseau de Petri déduit d'une modélisation OCPN des contraintes de l'application (figure 3.4).

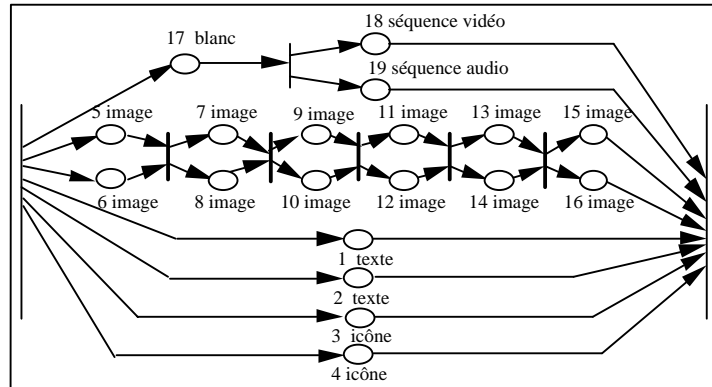


Figure 4.4. Réseau de Petri déduit du modèle OCPN de l'application considérée

On y constate que l'ordre de présentation des textes et des icônes est indifférent : on peut alors envisager toutes les séquences de présentations possibles depuis (1, 2, 3, 4) jusqu'à (4, 3, 2, 1). En revanche, les images fixes sont liées par un ordre partiel : 7 et 8 peuvent être présentées dans un ordre quelconque, mais ni l'une ni l'autre ne peuvent précéder 5 ou 6, ni succéder à 9 ou 10. De manière globale, ce réseau de Petri fait apparaître un très grand nombre de séquence de présentation possible des 19 objets, dont en particulier (1, 2, ..., 19), que l'on supposera être la séquence d'émission.

Si l'on confronte ce réseau de Petri aux garanties d'ordre d'un service de Transport classique :

- un service sans ordre (tel que celui fourni par UDP) ne garantit pas la délivrance des données 7 et 8 (par exemple), avant celle des données 9 et 10, et de ce fait s'avère inadapté à l'application ;
- les données multimédias ayant été émises dans leur ordre de numérotation (de 1 à 19), un service totalement ordonné (tel que celui fourni par TCP) garantit une délivrance correcte des données ; cependant, il ne tire aucun profit du parallélisme de l'application ;
- un service respectant l'ordre partiel de la figure 4.10 prend exactement en compte les contraintes de synchronisation logique de l'application ; en cela, une POC autorise le fournisseur du service d'ordre partiel à exploiter les caractéristiques de l'application, tout en garantissant la QoS souhaitée par l'utilisateur en termes d'ordre de délivrance des données.

Etudions à présent l'architecture de mise en œuvre des protocoles à ordre et à fiabilité partiels (section 5).

\$.5. Conception de la couche Transport à ordre et fiabilité partiels

Par nature, la mise en œuvre d'une application multimédia implique le transfert de différents flux de données (audio, vidéo, etc.), ayant chacun des caractéristiques et des contraintes spécifiques. De ce fait, un service de Transport « multimédia », c'est à dire adapté au transfert d'applications multimédias, doit fournir plusieurs ensembles de QoS, répondant chacun aux besoins d'un flux applicatif.

Dans ce contexte, les travaux de recherche poursuivis ces dernières années ont conduit à deux types résultats :

- l'extension du concept de QoS Transport [DANTHINE 94] [DIAZ 95], qui s'est par exemple traduite par la spécification de nouveaux paramètres de QoS à caractères temporels (délai de transit et gigue) et par la définition de nouvelles sémantiques de service indiquant l'action à entreprendre par le fournisseur du service en cas de dégradation de la QoS négociée ;
- la proposition de nouvelles architectures de Transport, répondant aux exigences différentes des flux applicatifs (audio, vidéo, etc.), mais ne prenant pas en compte les contraintes de synchronisations multimédias au niveau du système de communication proprement dit.

L'approche présentée ci-après est différente : sur la base d'une modélisation TSPN de l'application, l'architecture proposée repose sur une prise en compte des contraintes de synchronisation multimédias au niveau de la couche Transport [CHASSOT 96] [OWEZARSKI 98a]]. Plus précisément, cette architecture repose sur l'établissement puis la "coordination", au niveau Transport, de plusieurs POC "monomédias", chacune dédiée au transfert d'un flux et fournissant la qualité de service QoS_{*i*} requise pour le flux *i*.

Nous précisons maintenant les principes de cette architecture et analysons plus en détails son intérêt pour les applications multimédias.

\$.5.1. Conception générale

Afin de faciliter la compréhension des choix architecturaux et des mécanismes du protocole de Transport proposé, une application multimédia particulière reprise d'un exemple de [AMER 94] servira de fil conducteur à cette section.

Cette application consiste en la diffusion d'un bulletin météorologique sur deux fenêtres vidéo d'une station de travail distante (cf. Figure \$.5). La plus grande de ces fenêtres comporte les images relatives au présentateur du bulletin dont les propos sont délivrés par deux sorties audio. La plus petite des fenêtres, en haut à droite de

l'écran, fait apparaître les images d'une seconde personne qui traduit les paroles du présentateur en langage des signes.

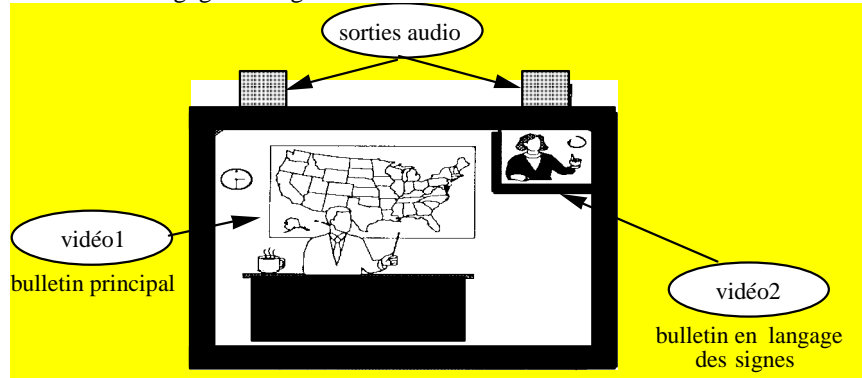


Figure \$.5. Un exemple d'application multimédia

Trois flux de données sont donc véhiculés dans cette application : deux pour les flux vidéo et un flux audio stéréophonique. Les caractéristiques de présentation de ces différents flux sont les suivantes :

- le flux relatif à la séquence vidéo centrale est composé de 30 images par seconde ;
- l'incrustation muette est animée à raison de 10 images par seconde ;
- le flux audio est composé de deux fois 60 échantillons par seconde.

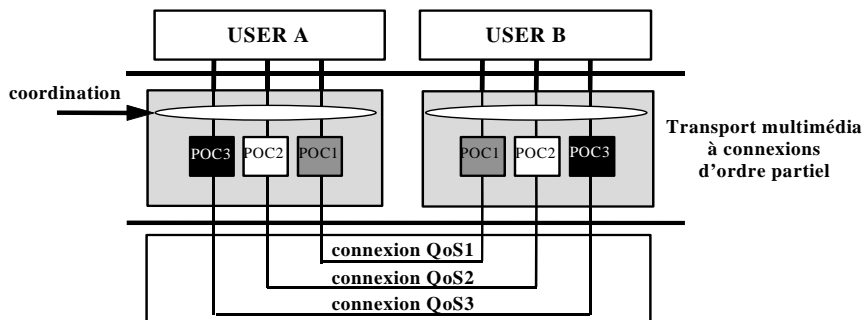


Figure \$.6 Architecture d'un Transport multimédia à connexions d'ordre partiel

Pour cette application, l'architecture de Transport multimédia envisagée (Figure \$.6) est composée de connexions de Transport monomédia, fournissant respectivement une QoS répondant aux caractéristiques du flux transporté (audio,

vidéo 1 et vidéo 2) mais, et à la différence des approches classiques, coordonnées entre elles pour préserver les relations logiques des objets multimédias. Nous appellerons ce Transport une multiconnexion multimédia à ordre partiel ou MM-POC.

La « coordination » est envisagée par le biais de deux paramètres : l'ordre partiel et la fiabilité partielle.

Gestion de l'ordre partiel

La gestion de l'ordre partiel se situe non seulement au travers de chaque POC monomédia, mais également entre ces connexions. En cela, elle permet de prendre en compte les contraintes de synchronisation logique à la fois intra et inter-flux déduites du modèle TSPN de l'application.

Une telle architecture permet ainsi de mettre en œuvre le service d'ordre partiel multimédia le plus adapté aux contraintes de synchronisation logique (c'est à dire hors aspect temporel) de l'application.

Gestion de la fiabilité partielle

L'intégration de la notion de fiabilité partielle repose sur un mécanisme de contrôle d'erreur pouvant être mis en œuvre soit « connexion par connexion », soit « par groupe de connexions ». L'objectif de chacun de ces mécanismes est de permettre la remise au plus tôt d'une donnée non « délivrable » au regard de l'ordre partiel (intra et inter-flux), au prix de la *déclaration de perte* de la(des) donnée(s) qui la précède(nt) dans l'ordre partiel. La déclaration de perte doit respecter la composante « fiabilité » de la QdS requise sur chaque POC.

La différence entre les deux mises en œuvre de ce mécanisme réside dans l'évaluation des déclarations de perte acceptables : avec un contrôle d'erreur « connexion par connexion », les données susceptibles d'être déclarées perdues pour permettre la délivrance d'une donnée *A* appartiennent nécessairement à la même connexion que *A*, cette restriction disparaissant pour un contrôle d'erreur « par groupe de connexions ».

Comme il sera montré en section (§.5.2), les deux variantes du mécanisme induisent une diminution globale du temps de transit des données applicatives avec une garantie de QdS en terme d'ordre et de fiabilité.

Analysons à présent de façon plus détaillée l'impact d'une gestion couplée de l'ordre et de la fiabilité au travers de l'architecture proposée.

§.5.2. Analyse du comportement intégré

Pour cela, considérons le réseau de Petri de la figure §.7 (dédié d'une modélisation TSPN de l'application décrite dans la Figure §.6), où chaque place représente une donnée utilisateur, que nous désignerons désormais sous le terme d'objet. Les places blanches et les places grises représentent des images entières.

Ce réseau de Petri, que nous qualifions d'ordre partiel multimédia P , traduit les contraintes de synchronisation logique de l'application pour une période de 0,1 seconde, et fait apparaître deux types de relations de dépendance :

- les relations de dépendance "intra-flux", telle que "l'objet 4 précède les objets 5 et 6 dans l'ordre partiel monomédia du flux audio" ;
- les relations de dépendance "inter-flux", telle que "l'objet 2 précède les objets 8 et 9 dans l'ordre partiel multimédia".

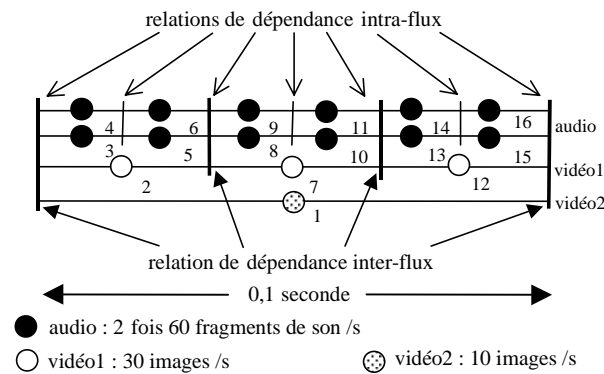


Figure §.7 Ordre partiel multimédia P et relations de dépendance intra et inter-flux

La prise en compte de P dans le cadre d'une architecture de Transport multimédia garantit la délivrance des objets selon l'une des séquences compatibles avec l'ordre P . En conséquence, l'intégration du concept de POC est perceptible à deux niveaux, *monomédia* et *multimédia*, respectivement liés à la gestion des relations de dépendance intra et inter-flux.

L'intégration "monomédia" du concept de POC consiste à respecter les relations de dépendances intra-flux dans chacune des connexions monomédia ; par exemple, la gestion des relations de dépendance intra-flux audio de la figure §.7 garantit la délivrance des objets 3 et 4 avant celle de 5 et 6, indépendamment de la délivrance des objets 1, 2, 7 et 12, relatives aux autres flux ; ainsi, les différentes connexions monomédias sont autant de connexions d'ordre partiel, l'utilisateur pouvant

sélectionner, pour chaque flux, l'ordre partiel souhaité parmi tous les ordres possibles.

L'intégration "multimédia" du concept de POC se manifeste au niveau de la "coordination". Elle a pour but de garantir la délivrance des objets en respectant les relations de dépendances inter-flux. Par exemple, en se référant à la figure \$.7, la délivrance des objets audio 8 et 9 n'est concevable qu'après délivrance de l'objet 2, du flux vidéo1.

L'intégration mono *et* multimédia du concept de POC permet donc de prendre en compte la première des caractéristiques d'un flux multimédia, l'ordre partiel. Cependant, l'architecture proposée en section \$.5.1 atteint toute son efficacité si l'on y intègre également la gestion de la fiabilité partielle, tirant ainsi parti d'une seconde caractéristique des flux multimédia : la sensibilité différente aux pertes et aux erreurs pour les médias constituant un flux multimédia. Ce second aspect est discuté maintenant .

Un transfert fiable au travers d'un médium imparfait implique la retransmission de toutes les données perdues et donc un accroissement de leur délai de transit de bout en bout. Ceci peut être inacceptable pour certaines applications temps réel telles que la visioconférence. En d'autres termes, les contraintes temporelles ne sont satisfaites qu'au prix d'une (éventuelle) dégradation de la fiabilité. Ceci étant, ces mêmes applications peuvent souvent supporter un transfert imparfait des données : la perte d'une image dans une séquence vidéo n'est par exemple pas perceptible par l'œil.

Il s'en suit que l'intégration de la notion de fiabilité partielle dans une POC permet une politique de délivrance "au plus tôt" des objets au prix d'une dégradation acceptable de la fiabilité (c'est à dire compatible avec la composante "fiabilité" de la QdS requise par l'utilisateur), tout en garantissant l'ordre partiel issu du modèle TSPN des contraintes de synchronisation de l'application. Le protocole offre ainsi la possibilité d'ignorer des objets précédant le dernier objet reçu, (qu'ils aient été perdus, retardés, ou altérés) : nous dirons que de tels objets sont *déclarés perdus*. Il s'agit alors de contrôler, lors de la délivrance de l'objet, la dégradation de la fiabilité consécutive à une éventuelle déclaration de perte en appliquant la règle suivante : *"la délivrance d'un objet rend obsolète tous les objets non encore délivrés ni déclarés perdus, qui le précèdent dans l'ordre partiel"*.

Dans l'exemple de l'ordre partiel multimédia de la figure \$.8, supposons l'objet 4 délivré et l'objet 5 parvenu à l'entité réceptrice tandis que l'objet 3 n'est pas encore reçu. Si l'objet 3 peut être déclaré perdu, l'objet 5 sera alors délivré à l'utilisateur sans délai.

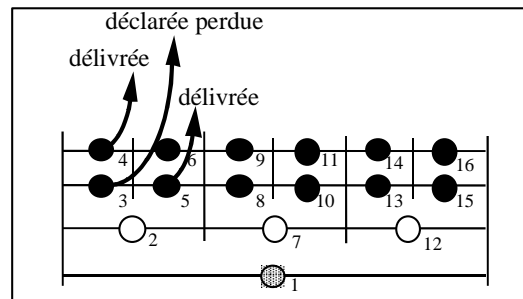


Figure \$.8 QdS "ordre"

A toute délivrance est ainsi attaché un coût en terme de fiabilité. La stratégie "au plus tôt" consiste à délivrer immédiatement tout objet dont le coût estimé de délivrance à l'utilisateur est inférieur au seuil permettant de garantir le taux de fiabilité² exigée par la QdS. La gestion de la fiabilité partielle peut être envisagée de deux façons :

- soit "connexion par connexion", ce qui impose que la délivrance d'un objet sur une POC_{*i*} n'affecte pas la fiabilité d'une POC_{*j*} avec $i \neq j$ (le coût de livraison d'un objet devient infini si elle rend obsolète un objet d'une autre POC) ;
- soit "par groupe de connexions", ce qui autorise la délivrance d'un objet sur une POC_{*i*} même si elle nécessite la déclaration de perte d'objet(s) d'une ou plusieurs POC_{*j*} avec $i \neq j$.

Dans l'exemple de l'ordre partiel multimédia de la figure \$.8, supposons l'objet 3 déclaré perdu et les objets 4 et 5 délivrés à l'utilisateur. Si à présent l'objet 8 parvient à l'entité transport réceptrice, celui ci est :

- stocké jusqu'à réception ou perte avérée³ de l'objet 2 dans le cas d'une gestion « connexion par connexion », l'objet 6 étant alors déclaré perdu⁴ ;

² Plusieurs expressions de la fiabilité peuvent être envisagées : nombre maximal de pertes consécutives, nombre maximal de pertes dans une période, nombre moyen de pertes, etc. A chacun de ces choix correspond une expression différente du coût et du seuil, mais l'algorithme de gestion de la fiabilité reste identique.

³ Cette perte pourra être avérée sur réception de l'objet 7.

⁴ Les objets déclarés perdus ne sont en aucun cas délivrés même si ils parviennent avec retard à l'entité Transport réceptrice.

– délivré sans attente à l'application tandis que les objets 2 et 6 sont déclarés perdus⁴ avec un mécanisme de gestion « par groupe de connexions ».

Notons que : a) l'objet 1, ne précédant pas l'objet 8 dans l'ordre partiel, n'est pas déclaré perdu par le protocole quelque soit le mécanisme mis en œuvre ; b) pour les deux mécanismes, les coûts liés aux déclarations de pertes sont supposés compatibles avec le paramètre de QoS « fiabilité partielle » de chaque POC.

Ce scénario illustre donc comment un protocole de Transport multimédia tire partie d'une dégradation acceptable de la fiabilité pour délivrer les objets au plus tôt en respectant l'ordre partiel, au prix d'une déclaration de perte des objets retardés ou effectivement perdus par le réseau. Dans les deux cas, le mécanisme de gestion de la fiabilité contribue à diminuer le délai de transit des données tout en garantissant conjointement les contraintes d'ordre et de fiabilité. Les deux options envisagées, de complexité équivalente, diffèrent par la priorité donnée aux performances (délai de transit) dans le cas d'une gestion « par groupe de connexions » ou à l'indépendance entre les flux dans le cas d'une gestion « connexion par connexion ».

§.5.3. Principes d'implantation

Une première évaluation de performances du protocole (en terme de temps de transit de bout en bout et de réservation de ressources mémoire) a été effectuée sur un prototype et enrichie par une campagne de simulation réalisée à l'aide de l'outil logiciel OPNET. Les résultats de cette étude [FOURNIER 97a], tout en confirmant l'intérêt du transport à ordre et fiabilité partiels, soulignent l'influence du temps aller-retour des PDU Transport (RTT : *Round Trip Time*), et du mécanisme d'acquiescement / retransmission sur les performances. Nous décrivons ici les points essentiels retenus dans [FOURNIER 97b] pour une implantation performante du Transport MM-POC.

Les hypothèses⁵ premières sur lesquelles repose l'implantation concernent les objets multimédias et le réseau sous-jacent : a) les données multimédias sont des flux continus et périodiques ; b) le réseau utilisé ne déséquence pas et présente un RTT stable. Bien que le choix d'ATM natif paraisse le plus adapté aux objectifs de

⁵ Ces restrictions répondent en fait à un grand nombre d'applications multimédias pour ce qui concerne les flux et correspondent par exemple aux technologies ATM et Ethernet pour le réseau. La relative stabilité des routes observée pendant la durée d'une association sur Internet permet de satisfaire « le plus souvent » l'hypothèse b).

ce type de protocole, l'implantation au-dessus de la pile UDP/IP a été retenue pour sa plus grande simplicité de réalisation et d'adaptation à d'autres technologies.

Au point de vue architectural, l'implantation représentée figure 5.9 diffère du modèle donné figure 5.6 par la présence d'une connexion TCP par laquelle transiteront les données de gestion de la connexion (signalisation). De même les POCs monomédias font appel à des associations UDP. Là encore, plutôt que de recourir au service fourni par IP, le choix s'est porté sur les protocoles de Transport UDP et TCP dans un souci de simplicité et d'efficacité (signalisation sûre grâce à TCP, multiplexage/démultiplexage délégué à UDP).

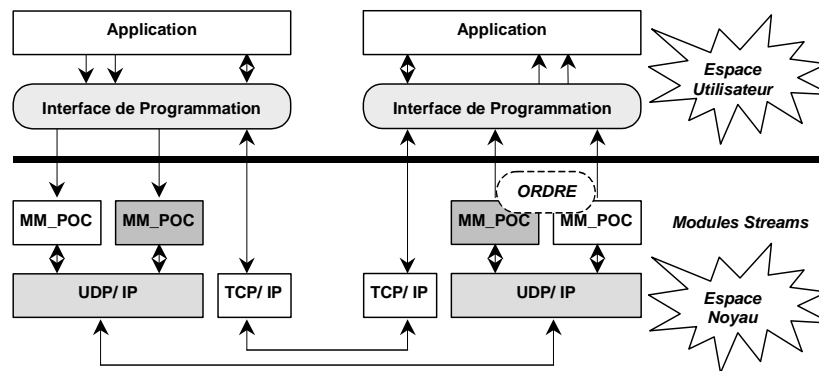


Figure 5.9 Architecture de l'implantation du Transport MM-POC

La figure 5.9 indique un choix supplémentaire constitué par l'implantation des POCs dans le noyau sous forme de module de *Stream*. Rappelons qu'un *Stream* [SUN 95] est un canal de communication bidirectionnel, permettant à un utilisateur de dialoguer de manière standardisée avec le pilote d'un périphérique ou d'un service réseau par exemple. Parmi les nombreux intérêts des *Streams*, l'implantation tire profit de la possibilité d'insérer dynamiquement un module de traitement entre la tête de *Stream* (qui assure l'interface avec l'utilisateur) et le pilote de service, en l'occurrence UDP (Pour plus de détails on consultera [FOURNIER 97b]). Les *Streams* offrent un cadre naturel pour l'écriture de piles de communication (gestion du parallélisme, code ré-entrant, optimisation des ressources mémoire, priorité de traitement, recopie des messages entre espace utilisateur et espace noyau, etc.)

5.5.4. Conclusion

Dans cette section, nous avons exposé les principes de conception et d'implantation d'une nouvelle architecture de Transport (MM-POC), utilisant et tirant parti du concept de connexion d'ordre partiel.

Comparativement à une approche basée sur l'utilisation d'un protocole de Transport tel que UDP, cette architecture décharge l'application de la gestion des retransmissions et de l'ordonnement des données en réception. En outre, le protocole MM-POC réduit le délai de transit de bout en bout et la taille des ressources mémoire nécessaires à la retransmission des données par rapport à un protocole tel que TCP. Cette augmentation de performances, constatée expérimentalement et par simulation, découle de mécanismes protocolaires qui tirent également partie des performances des réseaux à haut débit (ATM natif, IPv6, etc.) et des caractéristiques des flux multimédias.

Cependant, le protocole MM-POC ne peut garantir aux applications des contraintes temporelles de bout en bout : une telle garantie dépend de la qualité du service Réseau sur lequel il s'appuie et de la maîtrise des retards induits par l'asynchronisme des hôtes communicants. Nous présentons dans la section suivante comment ce dernier point peut être résolu au niveau applicatif en tirant parti des gains induits par utilisation du service MM-POC.

§.6. Application à une application de visioconférence

§.6.1. La philosophie de conception

Cette partie a pour but d'étudier et de développer une approche et un ensemble de mécanismes permettant de garantir aux utilisateurs des paramètres de Qualité de Service multimédia, comme la qualité audio, la qualité vidéo, le débit vidéo, le délai de bout en bout ou les contraintes de synchronisation temporelles, dans les applications multimédias distribuées. Plus particulièrement, cette partie va se focaliser sur la garantie des contraintes de synchronisation d'une application de visioconférence en environnement asynchrone (PNSVS : Petri Net Synchronized Videoconference System). En effet, la synchronisation multimédia est la contrainte la plus importante à assurer dans le cadre des systèmes distribués multimédias [BLAKOWSKI 95]. La problématique associée à la synchronisation multimédia consiste à assurer à la fois les contraintes de synchronisation intra et inter-flux. Garantir les contraintes de synchronisation intra-flux consiste à contrôler la gigue sur tous les objets (audio et vidéo dans le cadre d'une application de visioconférence) pour qu'elle reste en dessous d'une valeur maximale acceptable. Garantir la synchronisation inter-flux consiste à contrôler la dérive qui peut apparaître entre des flux parallèles (due à l'effet cumulatif de la jigue) et à la garder en dessous d'une valeur maximale (par exemple contrôler la dérive audio / vidéo

dans le cas de la synchronisation des lèvres dans une application de visioconférence).

La suite de cette partie a donc le plan suivant : tout d'abord, les contraintes de synchronisation d'une application comme PNSVS seront modélisées en utilisant le modèle TSPN (cf. partie §.3). Puis, il sera montré que le comportement de l'application de synchronisation peut être très différent de celui du scénario de synchronisation exprimé par l'utilisateur, pour tenir compte des caractéristiques spécifiques des composants matériels de l'ordinateur et du système opératoire. Cette partie (§.6.2.1) montrera en effet qu'il existe deux niveaux de synchronisation dans une application de visioconférence comme PNSVS. Puis, en analysant les résultats obtenus avec PNSVS, la partie (§.6.2.2) montrera qu'il est judicieux d'utiliser pour PNSVS un service transport à ordre partiel, qui permet par rapport à un transport standard de considérablement améliorer les performances et la qualité de présentation. La nouvelle architecture de PNSVS sera ainsi donnée, et elle sera évaluée, la partie (§.6.2.3) présentant les résultats d'évaluation.

A noter toutefois que PNSVS est un logiciel qui a été précédé dans le temps par l'application de visioconférence TSVS (Timestamp Synchronized Videoconference System) et qui utilisait une technique de synchronisation basée sur des estampilles temporelles. Toutefois, même si cette technique est simple et intuitive, elle comporte de nombreuses lacunes sémantiques qui la rendent peu adaptée au problème de la synchronisation multimédia :

- les estampilles ne peuvent pas formaliser les notions d'asynchronisme du système et de gigue tolérable sur les objets ;
- il est impossible au niveau du système de garantir des dates fixes de traitements à cause de l'asynchronisme du système ;
- les estampilles ne permettent qu'une connaissance à posteriori des contraintes de synchronisation à respecter, et elle ne peut donc pas anticiper les traitements à mettre en œuvre.

§.6.2. Système de visioconférence à QoS garantie

§.6.2.1. Qualité de service et modèle applicatif à deux niveaux

§.6.2.1.1. Utilisation des TSPN

Les estampilles ne représentent donc pas un modèle convenable pour modéliser des contraintes de synchronisation en environnement asynchrone, contrairement au modèle TSPN (voir partie §.3). Grâce à ses pouvoirs de modélisation et d'expression élevés, le modèle TSPN permet de modéliser facilement des scénarios de synchronisation de complexité forte. Le point essentiel de cette partie consiste à

étudier comment il est possible à partir d'un TSPN de décrire le comportement d'une couche de synchronisation, de façon à ce que pour tous les flux de données, les contraintes de synchronisation intra et inter-flux soient garanties.

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 \$.10 modélise une application de visioconférence dont les paramètres de synchronisation sont :

- débit : 10 images par seconde. Le temps de présentation nominal d'un objet est donc de 100ms ;

- gigue acceptable sur les objets audio et vidéo : 10 ms [JEFFAY 94]. Il s'en suit que les intervalles de validité temporelle sont [90, 100, 110] ;

- le son est le média le plus important : les synchronisations inter-flux sont de type « et-maître » sur le son. En effet, le son étant prépondérant par rapport à la vidéo, il se pose comme le flux maître et ses contraintes temporelles doivent être forcément respectées. De plus, l'attribut "et" de cette règle de synchronisation essaie autant que possible d'attendre et de respecter les contraintes sur le flux vidéo pour réduire le nombre de discontinuités ;

- la qualité de la synchronisation doit être excellente, c'est à dire que la dérive inter-flux ne doit pas excéder 100 ms [JEFFAY 94], 100 ms étant la limite en deçà de laquelle le décalage audio vidéo n'est pas perceptible par l'homme. 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.

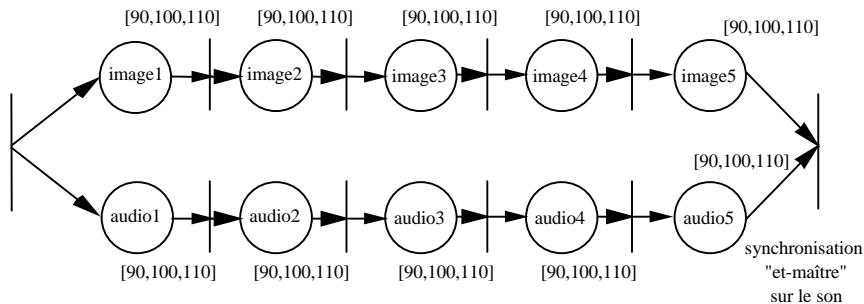


Figure \$.10 Exemple de TSPN pour la visioconférence à 10 images par seconde

\$.6.2.1.2. Latence des cartes multimédias et décalage des synchronisations inter-flux

Le TSPN de la figure \$.10 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, 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 \$.10, 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 $i+2$.

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 [OWEZARSKI 96a]). 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 $i-2$ (Figure \$.11), ce qui – après passage dans les cartes de présentation multimédia – donnera une présentation dans laquelle le son i et l'image i seront synchronisés.

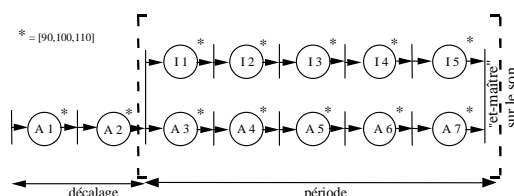


Figure \$.11 TSPN applicatif tenant compte des temps de latence matériels

\$.6.2.1.3. Les deux niveaux de modélisation de PNSVS

Dans ce qui précède apparaissent clairement deux niveaux de modélisation nécessaires à la modélisation des deux niveaux applicatifs. Le premier niveau correspond à l'interface dont le comportement est modélisé par le TSPN de présentation et qui correspond à la vue de l'utilisateur du scénario de synchronisation, c'est-à-dire à la façon dont les objets audio et vidéo sont synchronisés. Le TSPN de présentation est le même pour les deux utilisateurs : l'émetteur et le récepteur doivent manipuler les mêmes objets ayant les mêmes contraintes temporelles (cf. Figure \$.10).

Le second niveau correspond à la synchronisation applicative. La synchronisation applicative ne peut pas être modélisée en utilisant le TSPN de présentation, à cause des caractéristiques du système opératoire et des cartes multimédias. Ces caractéristiques obligent le moteur de synchronisation à considérer les objets multimédias d'une façon très différente de l'interface : le TSPN applicatif a donc été défini. De plus, l'émetteur et le récepteur sont différents et un modèle pour l'émetteur et un modèle pour le récepteur sont nécessaires. La figure \$.11 représente

le TSPN applicatif pour l'entité réceptrice de PNSVS⁶. [OWEZARSKI 96b] donne la définition complète des deux modèles TSPN, et montre comment les TSPN applicatifs sont automatiquement déduits du TSPN de présentation, qui est lui-même déduit de la QdS demandée par les utilisateurs.

\$.6.2.1.4. Architecture logicielle pour la synchronisation

Dans 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.

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 classes temps partagé et système. 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 d'ordonnancement s'appelle "Real Time" (RT). Toutefois, utiliser la classe d'ordonnancement 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. Les tâches temps réel doivent donc être courtes [OWEZARSKI 96b].

⁶ A noter toutefois que le modèle TSPN de la figure \$.17 peut encore subir des modifications. Déjà, les contraintes de synchronisation inter-flux peuvent amener à réduire la période de synchronisation. D'autre part, à cause du mode de fonctionnement des périphériques audio, on peut être amené à réduire la taille des paquets sons pour réduire le délai de bout en bout [OWEZARSKI 96b] [OWEZARSKI 98a]. D'ailleurs, sur le récapitulatif de la figure \$.21, il apparaît que la période de synchronisation a été réduite de 5 à 4 objets [OWEZARSKI 96b], et que les paquets son ont été découpés en deux pour réduire le délai de bout en bout (le temps de production des données audio étant ainsi divisé par 2, par conséquent, le délai de bout en bout l'est aussi).

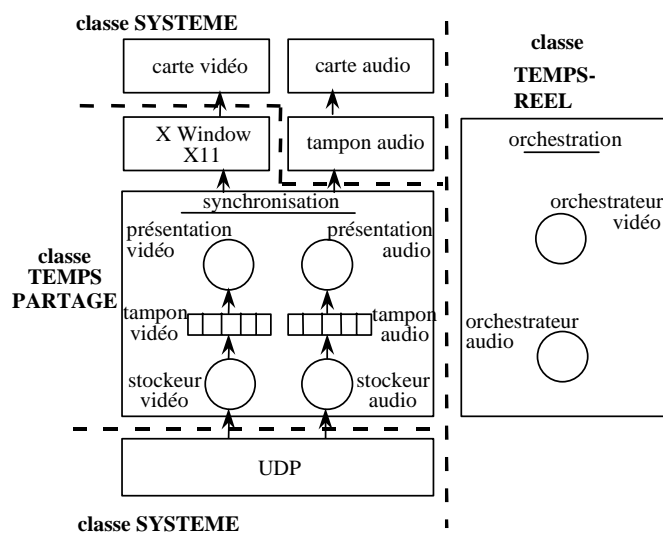


Figure §.12. Architecture de synchronisation pour la visioconférence

L'architecture de l'application de visioconférence résultant de ces problèmes est représentée sur la Figure §.12 [OWEZARSKI 95]. Elle représente les différents composants du système de visioconférence en associant à chacun d'eux leur classe d'ordonnancement 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 et à les conserver suffisamment longtemps pour résoudre les problèmes de gigue ;
- les processus de présentation audio et vidéo temps partagé qui réalisent les traitements nécessaires aux présentations des sons et des images vidéo ;
- les processus d'orchestration temps réel qui jouent le scénario de synchronisation modélisé par le TSPN applicatif récepteur et qui contrôlent les processus de présentation afin que ces derniers respectent les contraintes temporelles de présentation.

Avec ces processus, le principe de la synchronisation intra-flux repose sur un contrôle temps réel des processus de présentation par les processus d'orchestration. L'idée consiste ici à décorréler les tâches de présentation soumises à l'asynchronisme du système des tâches de contrôle temporel. Ensuite, la synchronisation inter-flux est mise en œuvre par l'intermédiaire d'un rendez vous entre les orchestrateurs audio

et vidéo qui respecte la sémantique de la transition « et-maître » du TSPN de la Figure \$.10.

\$.6.2.2. Analyse de l'architecture

Les principes de gestion de la QoS précédents ont été mis en œuvre dans le cadre de l'application de visioconférence PNSVS. Des mesures des temps de présentations des objets audio et vidéo de l'application ont montré que les contraintes de synchronisation étaient parfaitement respectées [OWEZARSKI 98a]. Cependant, il est apparu un problème de pertes excessives lorsqu'un réseau non fiable est utilisé. En fait chaque perte réseau engendre beaucoup plus qu'une simple discontinuité au niveau de l'interface de présentation. En effet, 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, comme les processus de présentation de PNSVS [OWEZARSKI 96b] 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, en cas de perte, 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. 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. 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.

Pour palier à ces problèmes, il faut disposer d'un mécanisme transport qui délivre les données et détecte les pertes au plus tôt.

\$.6.2.2.1. Une solution basée sur un transport à ordre partiel

Ceci ramène en fait à la présentation sur les protocoles à ordre partiel (partie \$.4), dont les mécanismes de gestion de l'ordre optimaux (notamment pour les flux audio et vidéo de la visioconférence) et surtout la notion de fiabilité partielle doivent fournir une solution adaptée pour les applications multimédias en général, et PNSVS en particulier. En effet, la notion de fiabilité partielle repose sur des principes de perte et délivrance au plus tôt. Ces mécanismes doivent donc permettre d'accélérer les communications de bout en bout tout en réduisant les besoins en mémoire et la charge du réseau.

La notion de fiabilité partielle 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. 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 manquants peuvent être perdus, alors ils seront considérés comme perdus, et l'objet reçu ne sera pas retardé (délivrance au plus tôt).

§.6.2.2.2. Architecture d'une application au dessus d'un transport à ordre partiel

L'architecture requise pour exécuter PNSVS au dessus d'un transport à ordre partiel est décrite sur la Figure §.13. Cette architecture ne positionne pas directement la tâche de synchronisation au dessus de l'ordre partiel car, à cause d'une gestion basée sur l'ordonnancement logique des données, le transport à ordre partiel 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 couche de pré-synchronisation a été ajoutée entre le transport et l'application pour apporter un contrôle temporel sur les données délivrées ou perdues par le Transport ; elle permet de détecter les longues séquences de pertes et les problèmes de dérive du réseau.

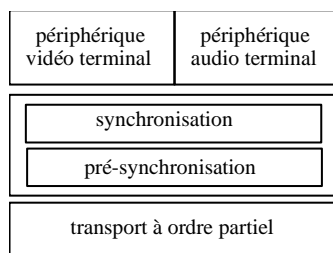


Figure §.13 *L'architecture de PNSVS au dessus d'un service transport à ordre partiel*

D'un point de vue processus l'architecture de PNSVS 2 lorsqu'un transport à ordre partiel est utilisé est très similaire à celle de PNSVS 1, si ce n'est l'apparition de deux processus de pré-synchronisation qui contrôlent le comportement temporel des stockeurs, de la même façon que les orchestrateurs contrôlent le comportement temporel des processus de présentation.

§.6.2.2.3. Modélisation des différents niveaux de synchronisation

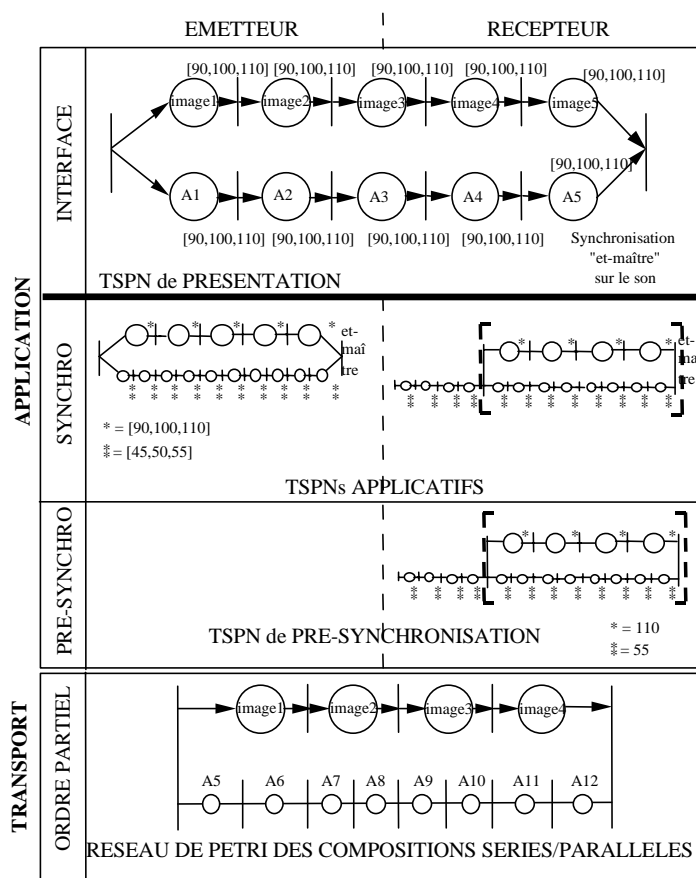


Figure §.14 Modélisation du comportement de chaque niveau de PNSVS

Il est à noter aussi que le modèle réseau de Petri apparaît pour modéliser les comportements de chacune des couches : interface utilisateur, application de synchronisation, transport, etc. et ceci avec des comportements différents pour

chaque couche. En fait, la figure \$.14 montre comment on peut modéliser, de proche en proche, le comportement de chacune des couches en fonction de leurs fonctionnalités et des contraintes qu'elles ont à respecter dans cette architecture : interface utilisateur, moteur de synchronisation émetteur et récepteur, moteur de pré-synchronisation et ordre partiel pris en compte par le transport.

\$.6.2.3. Evaluation

PNSVS a été implémentée sur des stations Sun (Sun Sparc Station 10, 5 ou 2) avec le système Solaris 2.5. Ces machines sont équipées de cartes vidéo Parallax qui permettent de capturer, afficher, compresser et décompresser des images utilisant le format de compression M-JPEG. La carte audio est la carte Sun standard livrée avec ce type de machines. Des tests ont été réalisés sur un réseau Ethernet à 10 Mbps et sur un réseau ATM à 155 Mbps.

PNSVS est une application de visioconférence qui peut traiter 25 images/s (320 x 240 pixels x 24 bits de codage des couleurs) bi-directionnel. Le délai de présentation minimum obtenu est d'environ 400 ms et ne peut guère être réduit à cause des temps de latence audio (environ 250 ms).

Il a été montré également que les contraintes temporelles de présentation audio et vidéo (gigue et dérive) étaient parfaitement vérifiées [OWEZARSKI 98a]. De plus, les bénéfices apportés par l'utilisation d'un service transport à ordre partiel sur une application comme PNSVS ont été évalués. Ainsi, la figure \$.15 présente, en fonction du taux de perte simulé sur le réseau de test, le taux de pertes supplémentaires engendrées par l'application, pour PNSVS 1 (qui utilise UDP) et PNSVS 2 (qui utilise un transport à ordre partiel). Les courbes montrent très largement le gain en qualité de service lorsqu'un transport à ordre partiel est utilisé.

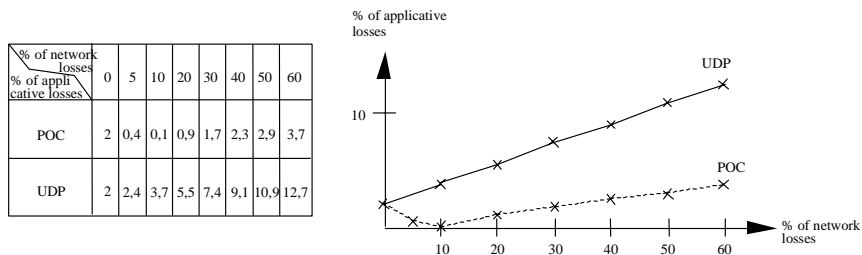


Figure \$.15 Evaluation comparative des pertes avec POC et UDP pour PNSVS

\$.6.3. Négociation et renégociation générique de Qualité de Service

En complément des fonctionnalités de garantie de la QoS, le besoin de pouvoir changer cette QoS à tout moment s'avère un besoin très fort pour les applications multimédias distribuées. En effet, une application utilisant une connexion (MM-)POC peut vouloir modifier, en cours d'utilisation, les paramètres de cette connexion : les relations d'ordre partiel sur les données et les paramètres de fiabilité. Ce besoin peut apparaître à la suite d'un changement des performances du réseau sous-jacent⁷, à un changement des besoins de l'utilisateur⁸, ou à un changement des besoins de l'application elle-même⁹. Cette modification de la QoS est en fait une modification de l'ordre partiel entre les données.

La solution simpliste à ce besoin serait de laisser à la charge de l'application l'ouverture d'une seconde connexion avec la nouvelle QoS désirée, puis de passer de l'une à l'autre. Cela nous semble inutilement coûteux en ressources (mémoire et réseau), et en développement pour les utilisateurs de connexions MM-POC.

Nous allons donc étudier les besoins en QoS dynamique pour les applications à MM-POC et l'architecture de communication qui en découle, puis sur l'exemple de PNSVS, il sera présenté un enrichissement de ce système de visioconférence pour intégrer des mécanismes de QoS dynamique.

\$.6.3.1 Les besoins et l'approche ordre et fiabilité partiels

\$.6.3.1.1 Besoins et contraintes

Etudions pour cela les besoins d'une application utilisant une connexion POC associés à une modification dynamique de QoS.

Nous nous plaçons dans le cas d'une application transportant un flux multimédia. Si l'application utilise une connexion MM-POC, c'est qu'elle a des besoins en terme de délais de bout en bout : il est important que les données ne soient pas ralenties inutilement par la connexion. Elle a aussi habituellement un besoin de débit continu : elle envoie des données de façon continue, et ne veut pas d'interruption du service¹⁰. De plus, une application qui utilise une connexion MM-POC attend que

⁷ Une augmentation du taux de perte peut conduire à vouloir augmenter le taux de perte acceptable, afin d'éviter les retransmissions et préserver le délais de bout en bout.

⁸ Plus attentif à cette partie du programme, il désire avoir une meilleure qualité de transmission.

⁹ qui peut passer de la présentation d'un flux MPEG sous-titré à celle d'un MJPEG

¹⁰ Ceci est surtout vrai dans les applications qui émettent des flux vivants (visioconférence, simulation temps réel) : aussitôt les données produites, il faut les émettre. Pour les application

les données lui soient délivrées dans un certain ordre. Lors de la modification de la QdS, on prendra bien garde à respecter ses trois impératifs de l'application : ne pas augmenter inutilement le délais de bout en bout, ne pas interrompre le flux, et continuer à garantir l'ordre.

Regardons maintenant les besoins spécifiques de la renégociation de QdS. Il faut pouvoir modifier les paramètres de la connexion MM-POC, c'est à dire, pouvoir changer l'ordre entre les données, la fiabilité intra et inter-flux, et, éventuellement, ouvrir ou fermer une ou plusieurs nouvelles connexions POC (apparition ou disparition de média). Comme nous le verrons, la difficulté n'est pas tant dans la renégociation, que dans la synchronisation du moment de changement de QdS.

Etudions donc maintenant ses besoins pour voir quelles contraintes ils imposent à notre architecture de renégociation de QdS.

\$.6.3.1.1 Organisation de la renégociation

L'ordre d'une connexion MM-POC est donné par un réseau de Petri, qui constitue le motif de base. Le flux transporté est une mise en séquence de ce motif de base. Lorsque l'on veut changer d'ordre, on va en fait passer d'un motif à un autre. Le flux transporté sera donc une séquence de motif initial mise en séquence avec une avec une séquence du nouveau motif. La *garantie de l'ordre* implique bien sûr que les données soient traitées en réception suivant la QdS avec laquelle elles ont été émises. De façon évidente, il faut que l'émetteur et le récepteur connaissent la nouvelle QdS. Il devront donc dialoguer. Pour se faire, il va falloir mettre en place deux agents de renégociation de la QdS dans le module MM-POC (qui utiliseront la connexion de contrôle). De plus, le changement de QdS ne peut pas se faire n'importe quand, mais doit se faire entre deux motifs.

Mais comment l'émetteur et le récepteur peuvent-ils s'accorder sur quel sera le dernier motif avec l'ancienne QdS et quel sera le premier avec la nouvelle ? Une solution simpliste consisterait pour l'émetteur à cesser l'émission après le dernier motif relevant de l'ancienne QdS, à envoyer un message de changement de QdS, à s'assurer qu'il a bien été reçu, puis à recommencer à émettre avec la nouvelle QdS. Cette solution est inacceptable car elle viole l'exigence de *continuité du service*. Une autre possibilité est pour l'émetteur et le récepteur de se mettre d'accord par avance

transmettant des flux mort (vidéo à la demande), on peut interrompre temporairement le flux, si le récepteur a assez de données dans ses tampons pour éviter la pénurie.

sur un moment de commutation¹¹, par une communication sur la connexion de contrôle. Le problème est que l'on n'est jamais sûr d'arriver à se mettre d'accord en un temps fini sur un réseau non fiable ou non borné (UDP ou TCP). En effet, si l'émetteur décide de commuter dans « n » motifs, il doit indiquer son choix au récepteur, mais n'est jamais sûr que son message arrive avant que le flux ne change de QdS. Si le récepteur envoie un message indiquant « n » et attend une confirmation, il se peut que le moment de commutation arrive sans qu'il ait reçu de confirmation : il ne sait alors pas si le récepteur n'a rien reçu (et qu'il ne doit pas commuter) ou a bien reçu « n » mais que l'accusé de réception est perdu (dans ce cas, il doit commuter). Ces exemples sont simplement là pour donner une idée du problème : plus formellement, c'est l'impossibilité du consensus sur réseau non borné qui est en cause. On peut vouloir contourner le problème de façon statistique : en choisissant assez à l'avance le moment de commutation, on est à peu près sûr que le message arrivera à temps. Cette solution cependant ne peut pas être retenue car, *primo*, elle néglige le fait que si on change de QdS, c'est peut-être justement que le réseau est engorgé, et que le message va avoir du mal à passer, *secundo*, elle introduit un temps relativement grand entre le temps où la demande de renégociation de QdS a lieu et le moment où elle est effective.

La solution retenue est en fait de faire un codage *in-band* du changement de QdS, c'est à dire d'ajouter aux données en transit un en-tête relatif à la QdS en cours. Nous pouvons nous permettre cet ajout car ceci peut être codé sur quelques bits, ce qui n'est pas grand chose pour un trafic multimédia. Bien sûr, ce codage ne remplace pas la négociation de la nouvelle QdS entre les deux agents. La renégociation a d'abord lieu, sur la connexion de contrôle, puis ensuite vient la commutation, codée avec les données.

Il va de soit que l'impératif de la commutation d'une QdS à une autre soit la plus rapide possible, afin de préserver le *délais de bout en bout*. Si la nouvelle QdS demande des ressources supplémentaires¹², elles doivent être réservées avant de faire la commutation. Ceci décompose donc la renégociation en deux phases distinctes : la *négociation* au cours de laquelle les deux agents vont négocier une nouvelle QdS et réserver les ressources associées, puis, la *commutation* où chaque entité va modifier sa QdS courante. Notons que pour que l'émetteur puisse commuter, il doit recevoir un message du récepteur lui signifiant que les ressources sont en place.

¹¹ En fait, le choix d'un moment de commutation se réduit au choix d'un indice, puisque les données sont numérotées.

¹² comme de la mémoire, ou l'ouverture d'une nouvelle connexion POC

§.6.3.1.3 Une commutation coordonnée

Etudions la commutation elle-même : elle doit en fait être prise en compte flux par flux, et non pas de façon globale. En effet, de part l'asynchronisme du traitement, les rafales du réseau, un module POC peut commencer à recevoir du réseau des données associées à la nouvelle QoS pendant qu'un autre POC délivre les dernières données associées à l'ancienne. Ils ne peuvent donc pas commuter exactement au même moment : chaque module doit donc gérer lui-même le moment de sa commutation.

§.6.3.2 L'architecture

Les contraintes énoncées précédemment donnent donc lieu à l'architecture présentée dans la figure §.16

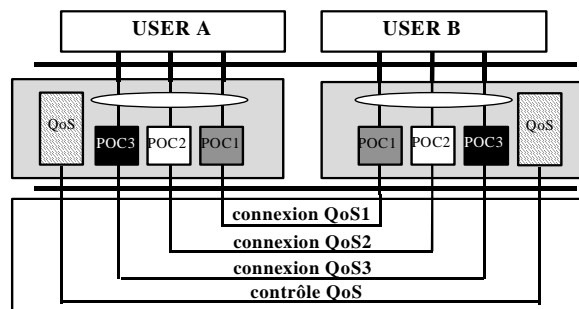


Figure §.16 Architecture MM-POC à QoS dynamique

Cette architecture n'est pas très différente de l'architecture MM-POC à QoS statique (figure §.6) : on a simplement introduit un agent de renégociation de QoS à chaque bout et une connexion fiable entre les deux agents (en fait, suivant l'implémentation du protocole, on peut réutiliser la connexion de contrôle – figure §.9 – ou ouvrir une seconde connexion, cela importe peu : l'important est d'avoir une connexion fiable, et distincte des connexions de données).

Cette architecture n'est pas très parlante sans présentation du protocole de renégociation de QoS associé. Comme nous l'avons déjà présenté, la renégociation se décompose en une *négociation* puis une *commutation*. La négociation est présentée sur la figure §.17. Elle commence par un dialogue entre les deux agents

pour définir la nouvelle QoS¹³, suivit de la réservation des ressources associées (en fait, ces deux étapes sont mêlées, puisque pour s'assurer que des ressources sont disponibles, un agent va généralement tenter de les réserver, puis regarder la réponse du système). Elle se termine par un message `RESSOURCES-OK` du récepteur vers l'émetteur, puis, lorsque le récepteur a enfin et reçu ce message et finit sa propre réservation, il peut, soit commuter au prochain motif, soit signaler à l'application qu'elle peut commuter.

La commutation elle, se fait de façon très simple : dès qu'un module reçoit des données associées à la nouvelle QoS, il commute de l'ancienne à la nouvelle QoS (on notera que, comme les motifs associés aux deux QoS sont strictement en séquence, aucun module ne peut recevoir des données de la nouvelle QoS alors qu'il lui reste à traiter des données de l'ancienne).

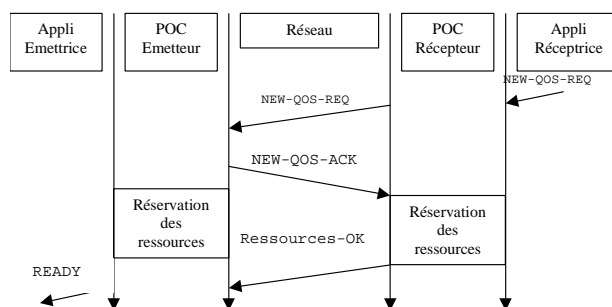


Figure §.17 Protocole de négociation

§.6.3.3. Application à la visioconférence

L'intégration de la renégociation de QoS dans l'application de visioconférence PNSVS a été réalisée suivant les mêmes contraintes que pour le protocole MM-POC. La solution retenue a donc été assez logiquement semblable à celle du module MM-POC : création d'un agent de renégociation de QoS¹⁴, et commutation par flux.

¹³ Le dialogue lui même peut prendre plusieurs formes : à l'initiative de l'un ou de l'autre, avec de multiples propositions ou une seule... Ce n'est pas le cœur du problème. Dans notre exemple, nous avons choisis de laisser le changement de QoS à l'initiative du récepteur, car c'est généralement lui qui a la meilleure vue de la QoS réellement fournie, mais on doit aussi laisser la possibilité à l'émetteur d'être initiateur de cette renégociation.

¹⁴ car, au niveau de l'application aussi, il faut tester la présence de ressources

La principale différence entre le module MM-POC et l'application PNSVS (du point de vue de la renégociation), vient du fait que l'application de visioconférence est constituée de deux couches (réception et présentation) et possède de contraintes temps réel. Le fait de comporter deux couches oblige à faire la commutation de chaque couche indépendamment : en effet, lorsque la réception voit arriver la première donnée avec la nouvelle QoS, il est fort probable que la couche de présentation soit encore en train de traiter des données associées à l'ancienne QoS. Donc, alors que dans MM-POC, l'unité de base de la commutation était un flux, dans l'application, on divise cette unité en deux sous couches.

L'implémentation de la renégociation dynamique de QoS, présentée dans [OWEZARSKI 97] a consisté à offrir à l'utilisateur la possibilité de modifier le nombre de sons et d'images par seconde traités par l'application (ce qui donne lieu à un changement de l'ordre entre les données, puisqu'une synchronisation inter-flux est réalisée chaque seconde). Cette implémentation a permis de valider les principes, par quelques mesures de performances [BOYER 98], [OWEZARSKI 98b].

§.7. Application à un transport de vidéo

La solution présentée dans le cadre de la visioconférence est une solution générique valable pour tous les médias et toutes les applications. Pour permettre cette généralité, elle n'utilise pas les caractéristiques intrinsèques du codage des données. En tenant compte, il est possible de concevoir des solutions spécifiques améliorant encore l'efficacité des protocoles de transport à ordre partiel. Pour cela, il faut pouvoir donner des informations supplémentaires sur la sémantique du codage des données transmises. La suite va montrer comment il est possible d'améliorer l'utilisation d'un transport de vidéo, notamment dans le cas d'un codage MPEG.

La création de normes internationales pour la compression vidéo, telles que H.261 [H261 93], MPEG [MPEG-1 93] [MPEG-2 93] et H.263 [H263 97], a accéléré le développement de nouvelles applications distribuées. Cette classe d'applications, et en particulier celles présentant un degré élevé d'interactivité, imposent au système de communication des contraintes particulières telles qu'une bande passante garantie, un temps de transit borné et une gigue contrôlée. Si ces contraintes ne sont pas satisfaites, l'application peut souffrir d'un fonctionnement dégradé.

Dans les réseaux de communication les plus répandus tels qu'Internet et ATM, les supports de la qualité de service réseau sont souvent déjà disponibles ou en phase de conception, mais ces techniques ne sont pas globalement déployées. Ainsi, le transfert de vidéo sur un service réseau sans garantie de service est encore très couramment utilisé.

Actuellement, les protocoles de Transport permettant la mise en œuvre d'application sur Internet sont essentiellement représentés par les protocoles classiques, TCP et UDP. Comme nous l'avons présenté aux paragraphes §.2 et §.3, ces protocoles ne tirent pas partie de la nature de données transmises ainsi que la disponibilité des ressources disponibles dans le réseau. Dans ce cadre, un service de Transport à ordre et fiabilité partielle, tel que celui présenté en section §.4 et §.5 (POC) peut être une réponse pratique à la mise en œuvre d'un tel service, d'autant plus efficace qu'il est en mesure d'agir en connaissant les attributs sémantiques des différentes données. Ici, ces données seront les trames d'un codage vidéo à techniques de compensation de mouvement : MPEG.

§.7.1. Besoins en communication de la vidéo codée avec des techniques de compensation de mouvement

Parmi les techniques de compression de la vidéo, celles exploitant les redondances spatiales et temporelles sont les techniques atteignant un gain de compression maximal. La *vidéo codée avec des techniques de compensation de mouvement* (VCTCM) appartient à cette classe de techniques. Le chapitre ?? présente la structure du train binaire d'une telle vidéo. Nous nous contenterons ici de rappeler que ce type de codage considère trois types d'images : des images I, qui contiennent suffisamment d'information pour être décodées seules, des images P, qui ne peuvent être décodées correctement que si on a à disposition l'image de type I ou P qui la précède, et des images B, qui nécessitent d'avoir les deux images de type I ou P qui la précèdent et la suivent, comme illustré par la Figure §.18. Rappelons aussi que chaque image, quelque soit son type, est découpée en tranche. La corruption ou la perte d'une tranche n'empêche pas de décoder le reste de l'image.

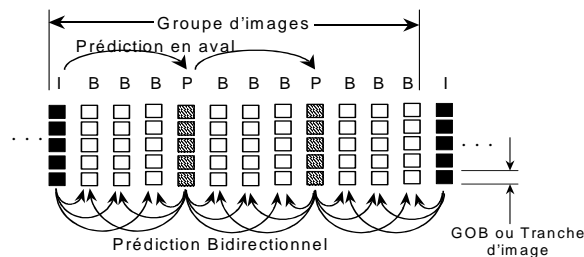


Figure §.18. Relations inter image dans la VCTCM

A la différence des applications informatiques classiques telles que le transfert d'un fichier (par exemple FTP, *File Transfert Protocol*), une application fondée sur le transfert de vidéo est capable de tolérer les erreurs introduites par le réseau. En

effet, la perte d'une image par seconde sur un flux à 25 images par seconde sera presque imperceptible. Par contre, la perte de toutes les images pendant 15 seconde sera bien moins tolérée par l'utilisateur. Ceci implique que le nombre et le type d'erreurs qu'une application est capable de tolérer doivent être contrôlés. Ceci constitue la *qualité de service* minimale imposée par l'application.

§.7.1.1. Pertes tolérées

D'un point de vue subjectif, lorsque la perte d'une donnée a un impact limité voire imperceptible, sur la qualité de présentation finale, on peut dire que cette perte est admissible. Dans cette section nous analysons la structure de la VCTCM afin de distinguer quels éléments peuvent être ainsi perdus. Une première approche montre qu'il existe deux types d'éléments dont l'importance diffère : les en-têtes et les données :

- *En-tête* : un en-tête contient les informations de contrôle indispensables au processus de décodage des données. Ce faisant, si un en-tête est perdu, le processus de décodage de données est sensiblement endommagé. Toutefois, la perte d'un en-tête est admis si la perte de la donnée associée est aussi admise.
- *Données* : de part leur rôle clé dans le décodage des images B d'un GOP, les données associées aux images de référence (I et P) sont considérées comme plus importantes que celles associées aux images B. La figure §.19 illustre les effets produits par la perte de données associées à une image de référence.

Nous observons ainsi que la perte partielle d'une donnée de référence altère le résultat du décodage des données dépendantes. C'est ainsi qu'une perte à priori légère au niveau d'une image de référence, induit une altération de la présentation d'un grand nombre d'images successives. Cette altération ne sera complètement terminée qu'à l'apparition de la prochaine image I (donnée indépendante). De plus, si les images contiennent des mouvements verticaux, les erreurs s'amplifient sur une zone importante des images. Le résultat visuel de la perte d'un élément d'une image de référence devient ainsi perceptible à l'œil humain.

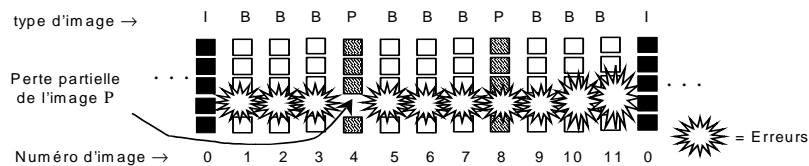


Figure §.19 Effets produits par la perte partielle d'une image de référence

§.7.1.2. Déséquences tolérés

Lorsque le décodage et l'affichage d'une donnée déséquentée ne présentent pas de dégradations perceptibles, on dit que cette donnée permet à l'application de tolérer des déséquentements. La tolérance au déséquentement de données est une propriété intéressante pour le système de communication car elle permet de relâcher des contraintes d'ordre sur certaines données transmises. Ainsi, un système de communication peut récupérer des données perdues pendant que d'autres données sont délivrées, en rupture avec l'ordre séquentiel d'émission. Le résultat est un service fiable avec un faible nombre de temps de blocage.

Afin de déterminer les éléments d'une VCTCM tolérant un traitement et un affichage déséquentés, une analyse doit être pratiquée sur la structure du train binaire. Les éléments de la VCTCM tolérant les déséquentements sont exclusivement les GOBs et les tranches d'image (TG) [HEYBEY 92] [TURLETTI 95][ROJAS-CARDENAS 98a][RHEE 98]. Pour des raisons de taille critique, nous nous intéresserons exclusivement aux possibilités de déséquentement des TG.

Si les TG tolèrent les déséquentements introduits par le réseau, ils doivent respecter un ordre partiel. En effet, les TG peuvent être délivrés en ordre quelconque, mais impérativement à l'intérieur des frontières de l'image à laquelle ils appartiennent. Nous nommerons ce type de contrainte le *déséquentement intra image*. Si cette règle n'est pas respectée, la qualité de présentation souffrira de dégradations notables. Ainsi, par exemple, dans la figure \$.19, les TG de l'image I (image 0) peuvent être délivrés dans un ordre quelconque, mais avant les TG de l'image B (image 1). Cette contrainte est nommée *ordre inter image*.

Pour définir l'ordre inter image des TG ainsi que le déséquentement intra image des TG, nous nous baserons sur l'entête des images constituant ici un élément de référence. Les en-têtes des images (par la suite nommés TT) doivent toujours rester ordonnés car ils déterminent l'ordre d'affichage des images. Les TG, quant à eux, n'ont qu'une seule contrainte : ils doivent se trouver après leur TT correspondant et avant le TT de l'image suivante.

\$.7.2. Unité de données pour le transport de MPEG

Compte tenu des besoins en communication exprimés précédemment, cette section propose la définition des unités de données de service niveau Transport (*TSDU : Transport Service Data Unit*) pour la VCTCM, en particulier pour la vidéo MPEG 1 [MPEG-1 93].

Les TSDU doivent être créés dans le but de permettre au récepteur (dans ce cas, le décodeur MPEG) de tolérer des pertes et des déséquentements de données. Les contraintes qu'un TSDU doit satisfaire afin d'atteindre ce but sont les suivantes :

1. Les éléments qui supportent de déséquences (TG) et ceux qui ne le supportent pas (TT), ne doivent pas être multiplexés dans une TSDU. Cette contrainte provient du fait que le mélange de ces éléments empêcherait d'exploiter la tolérance aux déséquences des TG.

2. Une TSDU ne doit contenir que des éléments (TT ou TG) entiers. Si un TT ou un TG est fragmenté et disséminé sur plusieurs TSDU, la perte ou le déséquence d'un seul TSDU rend impossible leur traitement.

3. La taille des TSDU doit correspondre à la taille maximale de l'unité de transmission MTU (Maximum Transmission Unit) du chemin émetteur/récepteur emprunté par les informations, ceci afin de tirer le meilleur profit de la capacité du canal de transmission [CLARK 90][MOGUL 90]. De plus, une taille de TSDU inférieure au MTU permet d'éviter la fragmentation des TSDU, opération qui dégrade de façon significative les performances du système de communication [KENT 87].

Ces considérations poussent à créer deux types de TSDU : le premier ne comportant que des TT, et le second ne comportant que des TG. Dans chaque cas, un ou plusieurs éléments entiers du même type peuvent être inclus dans un seul TSDU afin de profiter au mieux du canal de communication.

§.7.3. Spécification de l'ordre et de la fiabilité pour MPEG

Après avoir défini les caractéristiques des TSDU et les besoins en terme de communication imposés par le codage, nous abordons dans ce paragraphe la formalisation des relations d'ordre et de dépendance dans le cas d'un flux MPEG. Ces dernières seront modélisées par le biais du formalisme TSPN introduit dans la section §.3.

Dans la suite nous allons spécifier un cas particulier de séquence de vidéo MPEG. Cette séquence est caractérisée par les paramètres $N = 12$ (distance, en image, entre deux images I) et $M = 4$ (distance entre deux images de référence). En fait, les caractéristiques de cette séquence correspondent à celles de la séquence présentée dans la figure §.19 avec 15 tranches par image.

Considérons dans un premier temps la dimension d'ordre du service POC associé à notre exemple. Comme nous l'avons vu dans le paragraphe précédent, le formalisme TSPN permet l'expression de relations série et parallèles. La spécification du cas considéré doit prendre en compte les points suivants :

1. Les TSDU de type TT requièrent un service totalement ordonné. Ceci est spécifié par une composition série de places TSPN (figure §.20).

2. Les TSDU de type TG contenant l'information de référence (image de type I et P) supportent un déséquenceur intra-image et inter-image. A l'intérieur d'une trame, les TG peuvent être délivrés dans un ordre quelconque (déséquenceur intra-image). Pour modéliser cette caractéristique, une composition parallèle de places est utilisée. D'autre part, l'ordre inter-image des TG est limité en liant les places à la dernière transition MT12. Ceci permet d'indiquer que le protocole POC a la possibilité de retarder la livraison d'un TG jusqu'au tirage de la transition MT12. Notons que ces relations constituent un relâchement des contraintes de service, lequel rend possible la récupération de pertes sans blocage.

3. Les TSDU de type TG contenant de l'information relatives aux trames B ne supportent que le déséquenceur intra-image. A cet effet, une composition parallèle à validité limitée (par deux transitions contiguës) est spécifiée.

Deux caractéristiques importantes sont modélisées :

1. Les TSDU contenant des TG sont autorisées à être perdues.
2. Les TSDU de type TT ne doivent pas être perdus. Lorsque plusieurs objets dépendent d'un TT, on assigne au TT la catégorie d'objets maître et on relie tous les objets à une transition MASTER. En revanche, si aucun objet ne dépend de l'objet TT, une transition à entrée SIMPLE suffit. Les transitions de type MASTER sont représentées par le symbole MT_i alors que CT_i pour les transitions classiques.

Le TSPN modélisant les caractéristiques d'un flux MPEG est illustré figure §.20.

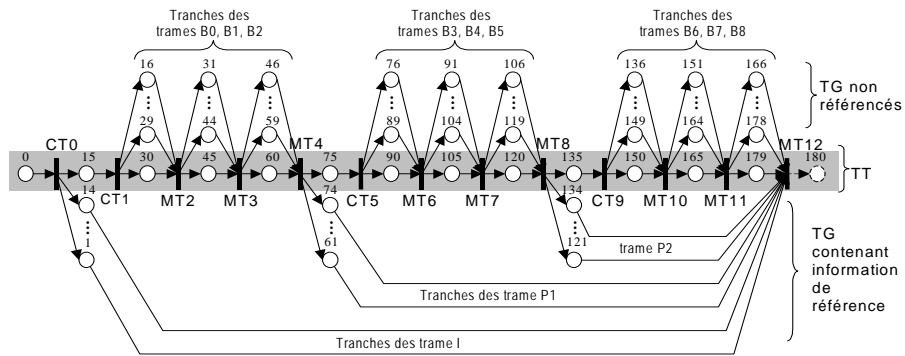


Figure §.20 Spécification TSPN du service POC pour une séquence de vidéo MPEG vidéo

§.7.4. Expérimentation

La stratégie de transport présentée a été mise en œuvre dans le contexte du réseau Internet. Des mesures en grandeur nature ont été réalisées sur la base de la

souche protocolaire réalisée et décrites dans [ROJAS-CARDENAS 99a], et d'une application de transfert vidéo MPEG.

La durée de la transmission est d'environ 11 secondes et le débit de présentation est de 25 images par seconde, une image étant constituée de 15 tranches et chaque tranche étant transportée par une TSDU.

Les expériences ont été menées dans la région Toulousaine, le serveur étant localisé au LAAS-CNRS et le client à l'ENSICA. Une dizaine de kilomètres et 5 routeurs séparent les deux sites. Le logiciel serveur fonctionne sur une station de travail Sun Ultra Sparc 1 et le récepteur sur une Ultra Sparc 2.

Les tests ont été effectués d'une part sur les empilements classiques TCP/IP et UDP/IP et d'autre part sur l'implantation du protocole proposée, soit POC/UDP/IP. Notons que l'objectif de ces expériences n'est pas de comparer les performances brutes de protocoles dont les services sont très différents, mais plutôt de mettre en évidence les avantages de l'ordre et de la fiabilité partiels (notamment au niveau de la continuité du service). Afin d'adapter le service TCP/IP aux besoins du transfert de vidéo, ce dernier a été utilisé en mode « sans attente » (TCP_NODELAY), pour éviter le comportement classique de TCP consistant à grouper plusieurs SDU dans une PDU qui introduit un délai supplémentaire.

§.7.4.1. Impact sur la qualité de présentation

L'impact du service fournit par le protocole POC a été testé grâce à l'utilisation d'un décodeur MPEG capable d'accepter des informations déséquilibrées. Les spécifications de ce décodeur sont plus largement décrites dans [ROJAS-CARDENAS 98a]. Les résultats visuels obtenus suite à l'utilisation des protocoles UDP et POC sont illustrés dans sur la figure §.21.

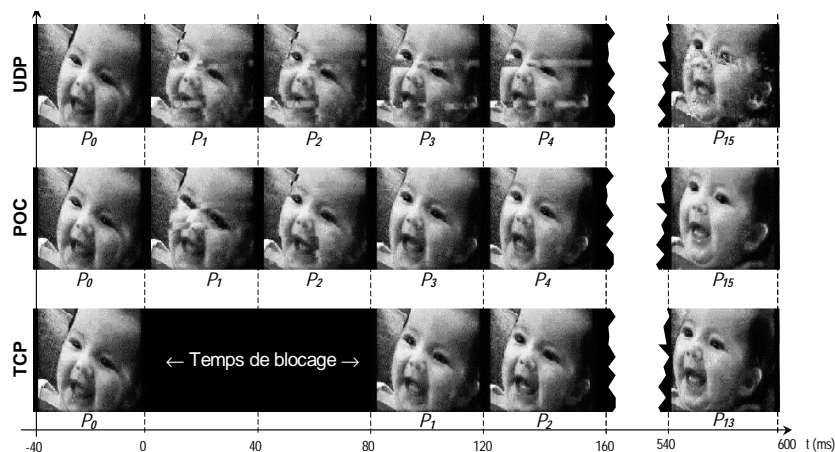
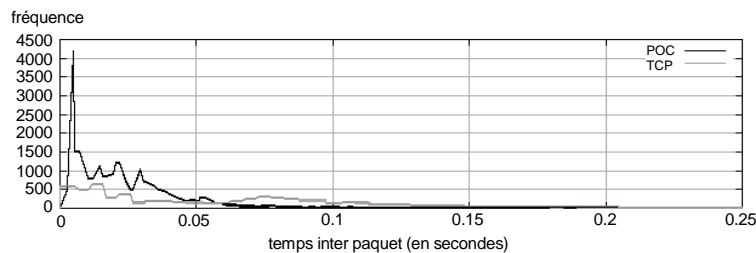


Figure \$.21 *UDP, TCP et POC pour le transport de vidéo MPEG*

Trois tranches ont été perdues lors de la transmission de l'image P1 de type I. Dans le cas d'UDP, outre la dégradation l'image I, les pertes ont entraîné l'altération de toutes les images jusqu'à l'image I suivante. Au contraire, l'utilisation du protocole POC limite cette altération aux deux d'images P1 et P2 correspondant au temps de récupération des trois tranches perdues. Notons que la perte des tranches ne provoque aucun blocage ni avec UDP, ni avec POC. Par contre, l'utilisation du protocole TCP, conduit à un blocage d'environ 80ms entre P0 et P1.

\$.7.4.2. Impact sur la continuité de service

Afin d'estimer l'impact d'un service à ordre et à fiabilité partiels sur la continuité du service nous utilisons TCP comme point de référence. Comme nous l'avons vu précédemment, une perte de TSDU entraîne le blocage d'une connexion TCP pendant le temps de la reprise de l'erreur. La figure suivante illustre la distribution de fréquence du temps inter paquet dans les conditions d'étude énoncées précédemment.

**Figure \$.22** *Distribution des fréquences de temps inter-paquets*

Observons dans la figure \$.22 que les fréquences de temps inter paquet d'une connexion TCP sont largement distribuées entre 0 et 200 ms. Au contraire, la distribution des fréquences pour une connexion d'ordre et de fiabilité partiels est notablement plus réduite et reste inférieure à 30ms. De plus, le temps inter paquet moyen pour une connexion TCP est de 63,8ms contre 20,6ms pour POC, très près de celle de UDP avec 20,3ms.

Plus d'informations sur ces travaux sont disponibles sur le site [POC WEB 99].

\$.8. Conclusion et perspectives

Les nouveaux services et protocoles multimédias se développent actuellement selon deux approches : l'une "Application orientée Réseau", qui considère que les applications doivent s'adapter aux problèmes du réseau ; la deuxième, "Réseau orienté Application", qui enrichit les réseaux actuels en y incluant toutes les caractéristiques nécessaires au transport des nouvelles applications.

Ce chapitre a développé la deuxième approche. Il a en particulier montré comment, à partir d'une modélisation formelle, non ambiguë et complète, des informations multimédias, un ensemble de nouveaux protocoles a été obtenu. De plus, ces protocoles définissent et font partie d'une architecture de communication originale.

Il a aussi été montré comment les résultats obtenus étendent conceptuellement les protocoles actuels, les protocoles orientés connexion, tel TCP, et les protocoles sans connexion, tel UDP, constituent deux cas particuliers de la solution proposée.

La nouvelle architecture a été développée et son comportement précisé. Les améliorations en termes de ressources informatiques nécessaires et les gains en termes de valeurs temporelles, à la fois pour les délais et l'interactivité, ont aussi été présentés.

Enfin, quelques exemples illustratifs importants liés à la gestion dynamique de connexion, au transfert des images vidéo, toujours en assurant une certaine qualité de service temporelle, ont été donnés.

Pour les prochaines années, il faut concevoir et proposer des systèmes distribués multimédias coopératifs qui permettront à un ensemble d'agents, partout dans le monde, de communiquer et de traiter les informations multimédias en temps réel.

Un tel but, à la fois très futuriste et très proche, nous paraît constituer un enjeu fondamental, mais les problèmes posés sont bien plus délicats à appréhender :

- l'architecture se complexifiera par les différentes contraintes liées aux nombreux participants,
- les besoins incluront différentes modalités, telles que les droits et les rôles.

En conséquence, ces systèmes d'une part posent de nouveaux problèmes dans de nombreux thèmes de recherche et d'autre part constituent un champ de recherche de toute première importance. Nous pensons que notre nouvelle approche peut être étendue afin de permettre une résolution élégante et performante de nombreux de ces problèmes.

Bibliographie

- [ALLEN 83] ALLEN J.F., "Maintaining knowledge about temporal intervals", *Communications of the ACM*, vol. 26, n° 11, November 1983.
- [AMER 94] AMER P. D., CHASSOT C., CONNOLLY C., CONRAD P. ET DIAZ M., "Partial Order Transport Service for Multimedia and other Applications". *IEEE/ACM Transactions on Networking*, vol.2, n° 5, Oct. 1994.
- [BERTHOMIEU 91] BERTHOMIEU B. et DIAZ M., "Modelling and verification of time dependant systems using time Petri nets", *IEEE transactions on software engineering*, vol. 17, n° 3, March 1991.
- [BLAKOWSKI 95] BLAKOWSKI G. ET STEINMETZ R., "A media synchronization survey : reference model, specification and case studies", *IEEE journal on selected areas in communications*, Vol. 14, No. 1, January 1996.
- [BOLOT 96] BOLOT J.C. ET TURLETTI T., "Adaptive error control for packet video in the Internet", Proc. ICIP '96, Lausanne, Sept. 1996.
- [BOLOT 98] BOLOT J.C. ET TURLETTI T., "Experience with rate control mechanisms for packet video in the Internet", *Computer Communication Review*, vol. 28, no.1, January 1998.
- [BOYER 98] BOYER M., OWEZARSKI P. ET DIAZ M., "Dynamic QoS Renegotiation in the PNSVS Videoconferencing Application", Proceedings of the Fifth International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS'98) , Oslo, Norway, September 1998.
- [CAMPBELL 94] CAMPBELL A., COULSON G. ET HUTCHINSON D., "A Quality of Service Architecture", *ACM Computer Communications Review*, April 1994.
- [CHASSOT 96] CHASSOT C., DIAZ M. ET LOZES A., "From the Partial Order Concept to Partial Order Multimedia Connections". *Journal for High Speed Networks*, vol.5, n°2, 1996.
- [CLARK 87] CLARK D.D., LAMBERT M.L. ET ZHANG L., "NETBLT: a bulk data transfer protocol". RFC 998, March 1987.
- [CLARK 90] CLARK D. D. AND TENENHOUSE D. L., "Architectural considerations for a new generation of protocols", in Proceedings ACM SIGCOMM, Philadelphia, September 1990.

- [CHERINTON 89] CHERINTON D.R. ET WILLIAMSON C.L., "VMTP as the Transport layer for high-performance distributed systems", *IEEE Communications Magazine*, pages 37-44, June 1989.
- [DANTHINE 94] DANTHINE A. Ed., "The OSI95 Transport Service with Multimedia Support", Research Reports ESPRIT, Project 5341, OSI95, Vol.1, Springer Verlag, 1994.
- [DELGROSSI 93] DELGROSSI L. ET SANDVOSS J., The BERKOM-II Multimedia Transport System, Aug. 1993.
- [DIAZ 93a] DIAZ M. ET SENAC P., "Time stream Petri nets, a model for multimedia streams synchronization", Proceedings of MultiMedia Modelling'93, Singapore, November 1993.
- [DIAZ 93b] DIAZ M., SENAC P. ET DE SAQUI-SANNES P., "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
- [DIAZ 94a] DIAZ M. ET PAYS G., "The Cesame Project : Formal design of high speed multimedia cooperative systems". *Annals of Telecommunications*, tome 49, n° 5-6, Mai-Juin 1994.
- [DIAZ 94b] DIAZ M., LOZES A., CHASSOT C. ET AMER P. D., "Partial Order Connections. A new concept for High Speed and Multimedia Services and Protocols", *Annales des Télécommunications*, tome 49, n°5-6, Mai-Juin 94.
- [DIAZ 95] DIAZ M., DRIRA K., LOZES A. ET CHASSOT M., "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.
- [DIOT 95a] DIOT C., HUITEMA C. ET TURLETTI T., "Network Conscious Applications". HPCS Workshop. Mystic. Aug 1995.
- [DIOT 95b] DIOT C., CHRISMONT I. ET RICHARDS A., "Application level framing and automated implementation". *IFIP High Performance Networking VI*, R. Puigjaner Ed., Chapman & Hall, p. 211-228, 1995.
- [FOURNIER 97a] FOURNIER M., CHASSOT C., LOZES A. ET DIAZ M., "Performance evaluation of partial order connections", 7th IFIP Conference on High Performance Networking, 1997.
- [FOURNIER 97b] FOURNIER M., Réalisation et évaluation de performances d'un protocole de transport multimédia à ordre partiel, Thèse de doctorat de l'Université Paul Sabatier de Toulouse, n°2618, 19 Mars 97.
- [H261 93] INTERNATIONAL TELECOMMUNICATION UNION, *Recommendation H.261: Video Coded for Audiovisual Services at p x 64 kbits*, International Telecommunication Union (ITU-T), 1993.
- [H263 97] INTERNATIONAL TELECOMMUNICATION UNION, *Recommendation H.263: Video Coding for Low Bit Rate Communication*, International Telecommunication Union (ITU-T), 1997.
- [HENCKEL 93] HENCKEL L., "Multimedia Communication Platform : Specification of the Enhanced Broadcast Transport Service", CIO RACE Project 2060, September 1993.
- [HEYBEY 92] HEYBEY A. T., Video Caging and the Application Level Framing Protocol Architecture, TR 542, Massachusetts Institute of Technology, 1992.

- [JEFFAY 94] JEFFAY K., STONE D. L., SMITH F. D., "Transport and display mechanisms for multimedia conferencing across packet-switched networks", *Computer networks and ISDN systems* Vol. 26, 1994.
- [KENT 87] KENT C. A., AND MOGUL J. C., "Fragmentation Considered Harmful", *Computer Communication Review*, vol. 17, no. 5, p. 390 – 401, Aug. 1987
- [LITTLE 90] LITTLE T. ET GHAFOR A., "Synchronization and Storage Models for Multimedia Objects". *IEEE J on Selected Areas in Communication*, vol 18, n° 3, p. 413-427, April 1990.
- [OWEZARSKI 95] OWEZARSKI P., DIAZ M. ET SENAC P., "Modélisation et implémentation de mécanismes de synchronisation multimédia dans une application de visioconférence", Actes du colloque francophone sur l'ingénierie des protocoles (CFIP'95), Rennes, France, 1995.
- [OWEZARSKI 96a] OWEZARSKI P. ET DIAZ M., "Models for enforcing multimedia synchronization in visioconference applications", Proceedings of the 3rd MultiMedia Modeling conference – Towards the information superhighway (MMM'96), World Scientific editor, Toulouse, France, 1996.
- [OWEZARSKI 96b] OWEZARSKI P., "Conception et formalisation d'une application de visioconférence coopérative. Application et extension pour la téléformation", thèse de doctorat, Université de Toulouse III, France, décembre, 1996.
- [OWEZARSKI 98a] OWEZARSKI P., DIAZ M. ET CHASSOT C., "A Time Efficient Architecture For Multimedia Applications", *IEEE Journal on Selected Areas in Communication JSAC.*, April 1998, vol.16, n°3, p. 383-396.
- [OWEZARSKI 97] OWEZARSKI P., BOYER M. ET DIAZ M., "Renégociation dynamique de qualité de service dans une application de visioconférence synchronisée", Actes du Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'97), 1997.
- [OWEZARSKI 98b] OWEZARSKI P., BOYER M., DIAZ M., "Mécanismes de gestion et de renégociation de la qualité de service dans une application de visioconférence", *Revue Electronique sur les Réseaux et l'Informatique Répartie (RERIR)*,(7), Septembre 1998.
- [MERLIN 74] MERLIN P., A study of the recoverability of computer systems, thesis, Computer Science dept., University of California, Irvine, 1974
- [MOGUL 90] MOGUL, J.C, AND DEERING, S. E., "Path MTU Discovery" Internet RFC 1191, 1990.
- [MPEG-1 93] CCITT, CCITT Recommendation MPEG-1, Coded Representation of Picture, Audio and Multimedia/Hypermedia Information, ISO/IEC 111172 Geneva Switzerland, 1993.
- [MPEG-2 93] CCITT, CCITT Recommendation MPEG-2, Coded Representation of Picture and Audio Information, ISO/IEC 13818/Draft, Geneva Switzerland, November 1993.
- [PEI 92] Protocol Engines Incorporated. "XTP protocol definition, Revision 3.6". January 1992.
- [POC WEB 99] POC World, <http://dmi.ensica.fr/poc>
- [RHEE 98] I. RHEE, "Error Control Techniques for Interactive Low-bit Rate Video Transmission", Proceedings of SIGCOMM, Vancouver, Canada, 1998.
- [ROJAS-CARDENAS 98a] ROJAS-CÁRDENAS L., CHAPUT E., DAIRAINÉ L., SENAC P., DIAZ M., "Video Transport Over Partial Order Connections", *HIPPARCH 98*, London, June 1998.

- [ROJAS-CARDENAS 98b] L. ROJAS-CÁRDENAS, P. SENAC, L. DAIRAINÉ et M. DIAZ, "Temporal Partial Order and Partial Reliability Service for Distributed Multimedia Applications", 5th International conference on Multimedia Modelling (MMM'98), Lausanne, Switzerland, Octobre 1998.
- [ROJAS-CARDENAS 99a] ROJAS-CÁRDENAS, CHAPUT E., DAIRAINÉ L., SENAC P. ET DIAZ M., "Video Transport Over Partial Order Connections", *Computer Networks*, vol. 31, Issue 7, Elsevier, April, 1999, p. 709-725.
- [ROJAS-CARDENAS 99b] L. ROJAS-CÁRDENAS, L. DAIRAINÉ, P. SENAC ET M. DIAZ, "An adaptive transport service for multimedia streams", IEEE International Conference on Multimedia Systems (ICMCS), Florence, Italy, Jun 1999.
- [ROJAS-CARDENAS 99c] ROJAS-CÁRDENAS L., DAIRAINÉ L., SÉNAC P. ET DIAZ M., "Towards a New Generation of Transport Services Adapted to Multimedia Applications", *Annals of Communications Journal, CNET*, 1999. (to appear)
- [SENAC 94] SENAC P., DIAZ M. ET DE SAQUI-SANNES P., "Toward a formal specification of multimedia synchronization scenarios". *Annales des télécommunications*, tome 49, n° 5-6, Mai-Juin 1994.
- [SENAC 96] SENAC P., DIAZ M., LEGER A. ET DE SAQUI-SANNES P., "Modeling logical and temporal synchronization in Hypermedia systems", *IEEE Journal on Selected Areas in Communications JSAC.*, 1996, p. 84-103.
- [SUN 95] SUN, "STREAMS Programmer's Guide", Unix System V Release 4, 1995.
- [TENET 94] TENET group. "Recent and Current Research", University of California, Berkeley, April, USA, 1994.
- [TURLETTI 95] T. TURLETTI, Contrôle de Transmission pour Logiciel de Vidéoconférence, Thèse de doctorat, <http://www-sop.inria.fr/rodeo/>, Apr. 1995.
- [WALTER 83] WALTER B., "Timed Petri-nets for modeling and analysing protocols with time", Proceedings of PSTV, III, IFIP, 1983
- [WILLRICH 96] WILLRICH R., DE SAQUI SANNES P., SENAC P. ET DIAZ M. "Hypermedia document design using the HTSPN model ", in Proceeding of 3rd International Conference on Multimedia Modeling (MMM'96), Toulouse (France), 12-15 Novembre 1996.